

---

# Herramienta de apoyo a la navegación web para personas con discapacidad

---



## MEMORIA TRABAJO DE FIN DE GRADO

Realizado por:  
Lorena Jiménez Corta

Dirigido por:  
Raquel Hervás Ballesteros  
Susana Bautista Blasco

Departamento de Ingeniería del Software e Inteligencia Artificial  
Facultad de Informática  
Universidad Complutense de Madrid

Junio 2018



# Herramienta de apoyo a la navegación web para personas con discapacidad

**Departamento de Ingeniería del Software e Inteligencia  
Artificial**

**Facultad de Informática  
Universidad Complutense de Madrid**

**Junio 2018**



# Agradecimientos

Agradezco al colegio Estudio 3 AFANIAS, en especial a los docentes que participaron, su colaboración y aportación con este proyecto.

A los alumnos del colegio Estudio 3 AFANIAS, por su participación, alegría, agradecimiento y buena acogida de este proyecto. Por enseñarme una visión del mundo totalmente diferente.

A mis tutoras, Raquel Hervás y Susana Bautista, por hacer posible que este proyecto sea de utilidad en un colegio de enseñanza. Por regalarme la experiencia de poder participar en un proyecto real.



# Frases

“Lo que sabemos es una gota de agua lo que ignoramos es el océano.”  
**Isaac Newton**

“Nuestros sentidos nos permiten percibir solo una pequeña porción del mundo exterior.”  
**Nikola Tesla**

“Solo una vida dedicada a los demás merece ser vivida.”  
**Albert Einstein**

“Creo que muchos niños se sienten solos y aislados en su propio mundo.”  
**Tim Burton**

“Las cosas que haces son las que te hacen, así que haz lo que sabes y sabrás lo que vales.”  
“Mientras veas sólo con tus ojos serás siempre ciego.”  
**El Chojin**

“Todo el mundo tiene el derecho a una vida digna y el deber de ayudar a los otros a tenerla cuando lo consigue.”  
**Juancho Marqués**





# Resumen

En la actualidad, cada vez más personas acceden a internet diariamente por diversos motivos. Sin embargo, no todos los usuarios poseen las mismas capacidades para navegar a través de internet. En el caso de los colectivos que presentan discapacidad o avanzada edad, entre otros, es habitual que encuentren diversas barreras que les dificultan el acceso, encontrándose con problemas para leer, escribir o entender textos a la hora de navegar por internet. Por ello, surge la necesidad de adaptar la web a estos colectivos, haciéndola accesible para todo tipo de usuarios. Una página web es accesible cuando sus contenidos están disponibles para todo tipo de usuario, permitiendo así interactuar de forma total e independiente con la página web. Existen determinadas pautas para hacer la web más accesible, por ejemplo, escoger un tipo de letra clara y de tamaño grande para facilitar la lectura del texto a personas con dificultad lectora, sin embargo, estas pautas no se cumplen en la mayoría de los casos.

Como proyecto de fin de grado se ha realizado el análisis, diseño e implementación de una herramienta de apoyo a la navegación web para colectivos de personas que presentan algún tipo de dificultad a la hora de comprender el lenguaje escrito. En todo el desarrollo del proyecto se ha llevado a cabo una aproximación centrada en el usuario final, con la colaboración del colegio Estudio 3 AFANIAS. Considerando las características de los usuarios que pueden tener problemas a la hora de leer o interpretar un texto, este proyecto se ha desarrollado con el fin de facilitar a dichos usuarios la navegación en internet, facilitándoles de manera sencilla un conjunto de herramientas que les ayuden a comprender los textos cuando estén navegando.

La aplicación, ReadIt, es una extensión desarrollada en Google Chrome, ya que es uno de los navegadores web más utilizados por la mayoría de los usuarios. A través del uso de servicios externos, permite a los usuarios comprender mejor el texto al que se enfrentan, por ejemplo, pudiendo buscar las distintas acepciones de una palabra que no entienden o mostrando un resumen del texto que proporciona la página web.

Para finalizar el proyecto, hemos probado nuestra aplicación con diez alumnos y profesionales del colegio Estudio 3 AFANIAS, obteniendo resultados realmente positivos y optimistas. Comprobamos la importancia que presenta nuestro proyecto, pudiendo observar de primera mano cómo las capacidades de comprensión lectora varían de una persona a otra. En el colegio Estudio 3 AFANIAS descubrimos que nuestra aplicación, ReadIt, no solo facilita la comprensión de los textos que ofrece internet, sino que también impulsa y anima a los usuarios a mejorar en su aprendizaje e independencia.

# Abstract

Currently, more and more people access the Internet daily for various reasons. However, not all users have the same abilities to navigate through the Internet. In the case of groups with disabilities and advanced age, among others, a barrier of difficulty to access this platform arises, encountering problems for reading, writing or understanding text when browsing through the Internet. Therefore, it is necessary to adapt the web to these groups, making it accessible to all users. A web page is accessible when its contents are available for all types of users, allowing total and independent interaction with the web page. There are certain guidelines to make the web more accessible. For example, choosing a clear font and large size to make the text easier to read for people with reading difficulties. However, this guidelines are not fulfilled in the most web pages. In this write, the analysis, design and implementation of a web browsing support tool for groups of people with written language understanding difficulties has been carried out, with a focus on end users thanks to the collaboration with the Estudio 3 Afanias school. Considering the characteristics of users who may have problems when reading or interpreting a text, this project has been developed in order to facilitate these users browsing Internet, providing them with a simple set of tools to help them understand the texts they encounter in web pages.

The application, called ReadIt, is an extension developed for Google Chrome, since it is currently one of the most used web browsers. Through the use of external services, ReadIt allows users to better understand the text they are trying to read. For example, it allows them to search for different meanings of a word they do not understand or display a summary of the text the web page provides.

To complete the project, ReadIt was tested with ten students and professionals from the Estudio 3 AFANIAS school, obtaining really positive and optimistic feedback. The importance of the project was then instantly apparent, being able to observe first-handedly how it assisted people with different reading comprehension skills. At Estudio 3 AFANIAS we discovered that ReadIt not only makes it easier for the users to understand the texts

offered by the Internet, but also promotes and encourages them to improve their learning and independence.

# Palabras Clave

- Accesibilidad web
- Lectura fácil
- Servicios web
- Extensión para Google Chrome
- API Chrome
- Diseño centrado en el usuario



# Keywords

- Web accessibility
- Easy to read
- Web services
- Google Chrome Extension
- API Chrome
- User-centered design





# Índice

<b>Agradecimientos</b>	<b>V</b>
<b>Frases</b>	<b>VII</b>
<b>Resumen</b>	<b>IX</b>
<b>Abstract</b>	<b>XI</b>
<b>Palabras Clave</b>	<b>XIII</b>
<b>Keywords</b>	<b>XV</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Objetivos . . . . .	2
1.2. Motivación . . . . .	2
1.3. Estructura del documento . . . . .	3
1.4. Gestión del proyecto . . . . .	4
<b>2. Preamble</b>	<b>5</b>
2.1. Goals . . . . .	6
2.2. Motivation . . . . .	6
2.3. Document structure . . . . .	7
2.4. Project management . . . . .	7
<b>3. Estado del arte</b>	<b>9</b>
3.1. El concepto de la accesibilidad web . . . . .	9
3.1.1. ¿Qué es la Accesibilidad Web? . . . . .	9
3.1.2. Pautas de Accesibilidad al Contenido en la Web . . . . .	10
3.1.3. ¿Por qué la Accesibilidad Web es importante? . . . . .	14
3.2. ¿Qué es la Lectura Fácil? . . . . .	14
3.2.1. Directrices internacionales de la IFLA . . . . .	14
3.2.2. ¿Por qué la lectura fácil es necesaria? . . . . .	17
3.3. Proyectos relacionados . . . . .	17

3.3.1.	inSuit . . . . .	17
3.3.2.	NavegaFácil . . . . .	18
3.3.3.	ATbar . . . . .	20
3.3.4.	Extensión WAVE Evaluation Tool . . . . .	23
3.3.5.	OpenDyslexic y OpenDyslexic Font-Helperbird-Free . . . . .	23
<b>4.</b>	<b>Tecnología</b>	<b>27</b>
4.1.	Orígenes de Google Chrome . . . . .	27
4.1.1.	Versiones . . . . .	28
4.2.	¿Qué es una extensión? . . . . .	29
4.2.1.	¿Por qué una extensión para Chrome? . . . . .	30
4.2.2.	Características básicas de Google Chrome . . . . .	31
4.3.	Estructura general de las extensiones en Google Chrome . . . . .	32
4.3.1.	Arquitectura de la ventana emergente . . . . .	34
4.4.	Mundos aislados en las extensiones de Google Chrome . . . . .	35
<b>5.</b>	<b>Diseño centrado en el usuario</b>	<b>39</b>
5.1.	Colegio Estudio 3 AFANIAS . . . . .	40
5.2.	Colegio Estudio 3 AFANIAS: Primera reunión . . . . .	40
5.2.1.	Mockup ejemplo 1: Click con el botón derecho. . . . .	41
5.2.2.	Mockup ejemplo 2: Barra de menú. . . . .	42
5.2.3.	Menú guardar. . . . .	45
5.2.4.	Marcas de guardado en el documento. . . . .	45
5.2.5.	Colegio Estudio 3 AFANIAS: Primeras conclusiones . . . . .	45
5.3.	Colegio Estudio 3 AFANIAS: Segunda reunión . . . . .	47
5.4.	Colegio Estudio 3 AFANIAS: Tercera reunión . . . . .	48
5.4.1.	Colegio Estudio 3 AFANIAS: Cuestionarios sobre la aplicación . . . . .	48
5.4.2.	Colegio Estudio 3 AFANIAS: Resultados tras la tercera reunión . . . . .	50
<b>6.</b>	<b>Descripción de la aplicación</b>	<b>55</b>
6.1.	Barra de menú de la aplicación . . . . .	55
6.1.1.	Barra de búsqueda . . . . .	55
6.1.2.	Blog . . . . .	56
6.1.3.	Definiciones . . . . .	56
6.1.4.	Palabras parecidas . . . . .	56
6.1.5.	Palabras contrarias . . . . .	57
6.1.6.	Pictogramas . . . . .	57
6.1.7.	Youtube . . . . .	57
6.1.8.	Wikipedia . . . . .	58
6.1.9.	Resumen . . . . .	58

---

6.1.10. Lectura en voz alta . . . . .	58
6.1.11. Icono de los engranajes . . . . .	58
6.2. Configuración de la barra de menú de la aplicación . . . . .	59
6.3. Guardar los datos del usuario . . . . .	60
6.4. Exportar los datos del usuario . . . . .	61
<b>7. Arquitectura de la aplicación</b>	<b>65</b>
7.1. Introducción . . . . .	65
7.2. Arquitectura . . . . .	66
7.2.1. Archivos principales . . . . .	66
7.2.2. SelectionServicesMenu . . . . .	67
7.2.3. Services . . . . .	68
7.2.4. Toolbar . . . . .	71
7.2.5. Modals . . . . .	71
7.2.6. Storage . . . . .	72
7.3. Instalación del plugin . . . . .	73
7.4. Extensibilidad del plugin . . . . .	74
<b>8. Conclusiones y trabajo futuro</b>	<b>79</b>
8.1. Conclusiones . . . . .	79
8.2. Trabajo futuro . . . . .	80
<b>9. Conclusions and future work</b>	<b>83</b>
9.1. Conclusions . . . . .	83
9.2. Future work . . . . .	84
<b>Bibliografía</b>	<b>85</b>



# Índice de figuras

3.1. Diseño de NavegaFácil . . . . .	19
3.2. Diseño de la barra de herramientas . . . . .	21
3.3. Resultados tras ejecutar WAVE . . . . .	24
3.4. Plugins OpenDyslexic y OpenDyslexic Font-Helperbird-Free . . . . .	25
3.5. Aplicando los plugins OpenDyslexic . . . . .	26
4.1. Versiones de Chrome ordenadas de menor a mayor estabilidad . . . . .	28
4.2. Uso de distintos tipos navegadores . . . . .	30
4.3. Estructura general de una extensión para Chrome . . . . .	32
4.4. Page Action vs Browser Action . . . . .	34
4.5. Arquitectura de la ventana emergente . . . . .	35
4.6. Ejemplo de una estructura DOM . . . . .	36
4.7. DOM compartido . . . . .	36
4.8. DOM compartido con mundos aislados . . . . .	37
4.9. Arquitectura que muestra el intercambio de información . . . . .	37
5.1. Selección de palabras pulsando botón derecho . . . . .	42
5.2. Selección de palabras a través de una barra de menú . . . . .	43
5.3. Selección de palabras a través de una barra de menú escribiendo texto . . . . .	44
5.4. Obteniendo los valores guardados por el usuario . . . . .	46
5.5. Ejemplo de marcas de guardado . . . . .	47
5.6. Respuestas en forma de semáforo . . . . .	49
6.1. Barra principal de la aplicación . . . . .	55
6.2. Definiciones de la palabra casa . . . . .	56
6.3. Resultado de una palabra que no presenta definiciones . . . . .	57
6.4. Pictogramas de la palabra casa . . . . .	58
6.5. Ejemplo usando la opción de resumen . . . . .	59
6.6. Ejemplo de los datos guardados por el usuario . . . . .	60
6.7. Resultado de filtrar los servicios adecuados para el usuario . . . . .	61
6.8. Guardando los datos del usuario . . . . .	62

6.9. Ejemplo de los datos guardados por el usuario . . . . .	63
6.10. Guardando los datos del usuario . . . . .	64
7.1. Estructura del proyecto . . . . .	66
7.2. Ruta para acceder a la pestaña <i>Extensiones</i> del navegador . .	74
7.3. Opciones que aparecen tras habilitar el <i>Modo desarrollador</i> .	74

# Índice de Tablas

5.1. Respuestas del cuestionarios de los alumnos . . . . .	52
5.2. Respuestas del cuestionarios de los profesores . . . . .	53





# Capítulo 1

## Introducción

La tecnología está cada día más arraigada en la sociedad actual de manera muy significativa. Con el paso del tiempo, cada vez existen más usuarios que acceden a internet a diario por diversos motivos.

Cabe suponer que no todos poseen las mismas capacidades para navegar en la web. Para muchos usuarios existe una gran barrera que les dificulta o incluso impide el uso parcial o total de la red. Esta barrera puede ser causada por distintos motivos: edad avanzada, personas extranjeras que no conocen el idioma, distintos tipos de discapacidades: visuales, motoras, sonoras, físicas, cognitivas...

Este proyecto nace con la idea de facilitar el acceso web a aquellos colectivos de personas que presentan algún tipo de dificultad a la hora de comprender el lenguaje escrito. Sin embargo, no todas las dificultades pueden tratarse de la misma forma, por lo que resulta realmente difícil abarcar todas en una sola aplicación.

Por tanto, el objetivo principal de este proyecto consiste en hacer la información web más accesible a aquellos colectivos de personas con discapacidades como Trastornos del Espectro Autista (TEA) o trastornos del lenguaje como afasia<sup>1</sup> o dislexia<sup>2</sup>, que tienen problemas con la lectura de los contenidos que se presentan en la web. Además, decidimos orientarlo a que cada usuario pudiera escoger la “ayuda” que necesita acorde a sus necesidades.

---

<sup>1</sup>Trastorno del lenguaje que se caracteriza por la incapacidad o la dificultad de comunicarse mediante el habla, la escritura o la mímica y se debe a lesiones cerebrales.

<sup>2</sup>Alteración de la capacidad de leer por la que se confunden o se altera el orden de letras, sílabas o palabras.

## 1.1. Objetivos

Este proyecto tiene como objetivo principal hacer la información web más accesible creando una herramienta de apoyo a la navegación adaptada a aquellas personas que presentan dificultades a la hora de comprender texto escrito.

Se pretende dar soporte al mayor número de las necesidades posibles, ya que cada persona es diferente y cada caso distinto, por lo tanto su tratamiento también debe ser diferente y totalmente personalizado.

La aplicación consiste en la creación de una extensión o *plugin* para Google Chrome que ofrezca distintas opciones para adaptarse a las distintas necesidades del usuario. La aplicación está implementada de tal manera que el profesor o la persona al cargo sea capaz de crear un perfil dependiendo de las necesidades del usuario y que éste sea capaz de buscar por sí mismo lo que necesite interactuando con la web.

## 1.2. Motivación

Durante seis años he sido profesora particular impartiendo clases a niños de primaria y secundaria, ayudándoles con las materias que les resultaban más complicadas, en especial las matemáticas. Siempre me han gustado los niños y saber que les ayudaba a aprobar sus materias más complicadas me llenaba de ilusión.

En el curso académico 2016/2107, me topé con una niña que presentaba déficit de atención. Tenía la capacidad suficiente para aprobar todas las materias con matrícula de honor, pero era incapaz de concentrarse. Esto supuso mayor empeño a la hora de impartir las clases y sobre todo, otra forma de explicar las materias. Al finalizar el curso, convertimos los suspensos en notables, pero éstos significaban más que una nota en un expediente académico. Eran una clara muestra de superación y esfuerzo.

En ese mismo periodo de tiempo, cursaba algunas asignaturas correspondientes con mi tercer y cuarto año en la carrera. No tenía nada claro qué era lo que quería hacer como trabajo de fin de grado y mucho menos a dónde dirigir mi futuro profesional, ya que nada me motivaba especialmente.

Cuando cursé la asignatura de Aplicaciones Web, descubrí que me apasionaba la programación web. Decidí hablar con mi profesora, que es mi actual tutora de trabajo de fin de grado, Raquel Hervás. Cuando me explicó la idea de este proyecto no dudé en querer participar. Iba a ayudar a un colectivo

de personas con dificultades a la par que desarrollaba una aplicación web.

No era consciente de todos los problemas de accesibilidad que se le presentaban a muchas personas, de hecho, ni siquiera me había planteado dicha problemática.

Sin duda, para mí es un orgullo saber que con los conocimientos adquiridos en la carrera durante estos años, podamos ayudar en alguna medida a personas con necesidades especiales.

### 1.3. Estructura del documento

Para la realización de este trabajo era importante estudiar el contexto sobre el que íbamos a desarrollar nuestra aplicación, por lo que comenzamos hablando en el capítulo 3 sobre los conceptos de la accesibilidad web y la lectura fácil, nombrando algunas de las aplicaciones que existen actualmente para conseguir que los sitios web sean más accesibles de una u otra manera.

Para continuar, decidimos tratar la tecnología escogida para desarrollar el proyecto en el capítulo 4, donde explicamos porqué decidimos desarrollar el proyecto en Google Chrome, estudiando la estructura general de sus extensiones.

En mitad del documento encontramos el capítulo 5, en el que redactamos nuestra experiencia trabajando con usuarios finales en el colegio Estudio 3 AFANIAS, mostrando los resultados obtenidos. Además, explicamos la importancia de realizar pruebas con los usuarios finales para poder desarrollar una aplicación que realmente sea de ayuda.

Posteriormente, en el capítulo 6 explicamos las distintas funcionalidades que podremos encontrar en nuestra aplicación, detallando el funcionamiento de cada una.

Acercándonos al final del documento, encontramos el capítulo 7 donde hablamos sobre la arquitectura de la aplicación. En primer lugar, explicando a alto nivel cómo están ordenados los ficheros de nuestra aplicación. En segundo lugar, desarrollamos en profundidad su contenido y funcionalidad.

El documento finaliza con el capítulo 8, donde presentamos las conclusiones y el trabajo futuro que se podría realizar al terminar el proyecto.

## 1.4. Gestión del proyecto

Para la realización del proyecto se han empleado las siguientes herramientas:

- Para la implementación del proyecto se ha hecho uso de *Atom*, un editor de texto multiplataforma y código abierto que puede utilizarse como IDE<sup>3</sup> para muchos lenguajes de programación.
- A la hora de desarrollar la memoria se ha hecho uso de *TeXstudio*, un editor de LaTeX<sup>4</sup> de código abierto y multiplataforma, trabajando sobre la plantilla *TeXiS*<sup>5</sup>.
- Para la gestión y control sobre los cambios efectuados en el código se ha utilizado *GitNIL*<sup>6</sup>, un repositorio dentro de la organización del grupo de investigación NIL<sup>7</sup> de la UCM en *Github*.

---

<sup>3</sup>Integrated Development Environment, es una aplicación informática que proporciona servicios integrales para facilitarle al programador el desarrollo de software.

<sup>4</sup>Es un sistema de composición de textos, orientado a la creación de documentos escritos que presenten una alta calidad tipográfica.

<sup>5</sup>Más información sobre TeXiS en <http://gaia.fdi.ucm.es/research/texis>

<sup>6</sup>Podemos encontrar el repositorio en <https://github.com/NILGroup/TFG-1718-AccesibilidadWeb>

<sup>7</sup>Más información sen <http://nil.fdi.ucm.es/>

## Capítulo 2

# Preamble

The technology is increasingly interwoven in the current society in a very significant way. Over time, there are more and more users accessing the Internet on a daily basis for various reasons.

It can be assumed that not everyone has the same ability to browse the web. There is a big barrier that makes it difficult or even impossible for many users to use partial or total network usage. This barrier can be caused by different reasons: advanced age, lack of understanding of the language by foreign people or different types of disabilities: visual, hearing, physical, cognitive...

This project was inspired by the idea of providing web access to these groups of people who have some type of difficulty understanding written language. However, not all difficulties can be dealt with in the same way, so it is really difficult to cover them all in a single application.

Therefore, the main objective of this project is to make the web information more accessible to groups of people with disabilities such as Autism Spectrum Disorders (ASD) or language disorders such as aphasia<sup>1</sup> or dyslexia<sup>2</sup> who have problems with reading the content presented on the website. In addition, it was oriented so each user could choose the “help” they need according to their needs.

---

<sup>1</sup>Language disorder, caused by brain injuries, characterized by the inability or difficulty to communicate through speech, writing or mimicry.

<sup>2</sup>Alteration of the ability to read which confuses or alters the order of letters, syllables or words.

## 2.1. Goals

The main objective of this project is to make web information more accessible by creating a navigation support tool adapted to people who have difficulties understanding written text. It is intended to support all needs. Since each case is different, treatment must also be different and fully personalized.

The application is an extension or *plugin* for Google Chrome that offers different options to adapt to different user's needs. The application is implemented in such a way that the teacher or person in charge is able to create a profile depending on the user's needs and then the user is able to search for his own needs by interacting with the web.

## 2.2. Motivation

For six years I have been a particular tutor teaching primary and secondary school children, helping them with the most difficult subjects, especially mathematics. I have always liked children and, knowing that I helped them pass their most complicated subjects, filled me with illusion.

In the 2016/2017 academic year, I came across with a girl with an attention deficit disorder. She had the intelligence to pass all the subjects with honors, but was unable to concentrate. This meant more effort in teaching and, more importantly, another way of explaining the subjects. At the end of the course, we turned the Fs into Bs. But that meant more than a grade as they were a clear sign of self-improvement and effort.

At the same period of time, I was taking some subjects corresponding to my third and fourth year of studies. I was not at all sure what I wanted to do as an bachelor's thesis, let alone where to direct my professional future, as nothing particularly motivated me.

When I took the Web Applications course, I discovered that I was passionate about web programming. I decided to talk to my teacher, who is my current work tutor, Raquel Hervás. When she explained the idea of this project to me, I did not hesitate to participate. I was going to help a group of people in difficulty while developing a web application!

I was not aware of all the accessibility problems that many people faces in their everyday life. In fact, I had not even thought about the issue at all.

Without a doubt, it is a source of pride for me to know that with the

knowledge acquired in my studies over the years, we can help people with special needs at least to some extent.

## 2.3. Document structure

For the implementation of this write, it was important to study the context in which we were going to develop our application, so we begin by discussing the concepts of Web Accessibility and Easy Reading in chapter 3, naming some of the applications that currently exist to make web sites more accessible in one way or another.

To continue, we discuss the technology chosen to develop the project in the chapter 4, where we explain why it was decided to develop the project in Google Chrome, studying the general structure of its extensions.

In the middle of the document we will find chapter 5, in which we write about our experience working with end users in the Estudio 3 AFANIAS school, showing the results obtained. In addition, we will explain the importance of testing with end users in order to develop an application that could really help them.

Subsequently, in chapter 6, we explain the different functionalities that can be found in our application, detailing the operation of each one.

Approaching the end of the document, we find chapter 7 where we will talk about the application architecture. First of all, we explain at a high level how the files of our application are arranged. Secondly, we develop its content and functionality in depth.

The document ends with chapter 8, where we will develop the conclusions and the future work that could be done at the end of the project.

## 2.4. Project management

The following tools have been used to carry out the project:

- *Atom* was used for the implementation of the project. Atom is a multi-platform open source text editor that can be used as IDE<sup>3</sup> for many programming languages.

---

<sup>3</sup>Integrated Development Environment, a software application that provides comprehensive services to facilitate the programmer the development of software.

- *TeXstudio* was used for the development of the memory. TeXstudio is an editor of LaTeX<sup>4</sup> open source and cross-platform, working on the *TeXiS*<sup>5</sup>.
- *GitNIL*<sup>6</sup>, a repository within the organization of the UCM's NIL<sup>7</sup> research group in Github, was used for code change management and control.

---

<sup>4</sup>It is a text composition system, aimed at the creation of written documents with a high typographic quality.

<sup>5</sup>Further information about TeXiS at <http://gaia.fdi.ucm.es/research/texis>

<sup>6</sup>We can find the repository at <https://github.com/NILGroup/TFG-1718-AccesibilidadWeb>

<sup>7</sup>More information about NIL can be found at <http://nil.fdi.ucm.es/>



## Capítulo 3

# Estado del arte

En este capítulo trataremos el tema de la accesibilidad web y la lectura fácil, así como las aplicaciones que existen en la actualidad para conseguir que los sitios web sean más accesibles para todos aquellos usuarios que lo necesiten.

### 3.1. El concepto de la accesibilidad web

El objetivo de esta sección es explicar el concepto de la “Accesibilidad web” y presentar las “pautas de Accesibilidad de Contenido” desarrolladas por el W3C<sup>1</sup> (W3C, 2018) y por qué son importantes.

#### 3.1.1. ¿Qué es la Accesibilidad Web?

La Accesibilidad Web (Mora, 2018) tiene como objetivo lograr que las páginas web sean utilizables por el máximo número de personas, independientemente de sus conocimientos o capacidades personales. Concretamente, ofrece la posibilidad de que la información web pueda ser comprendida y consultada por personas con discapacidad, cuyo objetivo es garantizar la igualdad de oportunidades, evitando así todo tipo de discriminación. Cuando hablamos de accesibilidad web se hace referencia a un diseño web que permitirá que estas personas puedan percibir, entender, navegar e interactuar con la web. Existen otros colectivos, como personas de avanzada edad a las que también beneficia la accesibilidad web.

La Accesibilidad Web engloba muchos tipos de discapacidades, incluyendo problemas visuales, auditivos, físicos, cognitivos, neurológicos y del

---

<sup>1</sup>El Consorcio World Wide Web (World Wide Web Consortium, en inglés) es una comunidad internacional donde diversas organizaciones y personas trabajan conjuntamente para desarrollar estándares web.

habla. Hoy en día, existen muchas personas con distintas discapacidades que no pueden utilizar la web, ya que actualmente la gran mayoría de los sitios web presentan grandes barreras de accesibilidad.

Para que el acceso a la información sea posible existen determinadas normas y requisitos que las páginas web deben cumplir, tal como indica la “Ley de Accesibilidad de la Información en las Páginas Web” (Ley N° 26.653). El cumplimiento de estas pautas contribuye a que el contenido sea más accesible para todas las personas, con o sin discapacidad, incluyendo a aquellos usuarios que para acceder a la web utilicen herramientas de apoyo (visuales, auditivas, motrices...).

### **3.1.2. Pautas de Accesibilidad al Contenido en la Web**

Una de las principales iniciativas del W3C es el desarrollo de normas de accesibilidad. El objetivo de la Iniciativa para la Accesibilidad Web (Web Accessibility Initiative, WAI (Henry, 2018)) es desarrollar los estándares de accesibilidad. Los grupos de trabajo del WAI desarrollan las normas de accesibilidad para los navegadores web, para las herramientas de autor, de evaluación y para el contenido web. Las normas del Grupo de Trabajo para el Contenido Web se llaman Pautas de Accesibilidad al Contenido en la Web (Web Content Accessibility Guidelines, WCAG).

Actualmente, existen dos versiones para las Pautas de Accesibilidad al Contenido en la Web.

#### **3.1.2.1. Pautas de Accesibilidad al Contenido en la Web 1.0**

La versión 1.0 de las Pautas de Accesibilidad al Contenido en la Web (Web Content Accessibility Guidelines 1.0, WCAG 1.0 (Chisholm et al., 1999)) fue un avance importante para lograr que Internet sea más accesible para las personas con discapacidad.

Estas pautas fueron escritas en 1999, creando 14 directrices y numerosos puntos de control que podían utilizarse para determinar la accesibilidad de una página web. Proporcionaban tres prioridades, niveles de cumplimiento o niveles de adecuación, es decir, la medida en que una página web cumple las directrices.

- PRIORIDAD 1 O NIVEL DE ADECUACIÓN A. Se trata de un requisito básico para que algunos grupos de usuarios pudieran usar el contenido web.
- PRIORIDAD 2 O NIVEL DE ADECUACIÓN AA. Indicaba una mejor

accesibilidad y la eliminación de importantes barreras de acceso al contenido.

- PRIORIDAD 3 O NIVEL DE ADECUACIÓN AAA. Proporcionaba mejoras a la accesibilidad del contenido web.

A continuación, desarrollamos las pautas de accesibilidad al contenido en la Web 1.0(Chisholm et al., 1999):

- Proporcionar alternativas equivalentes para el contenido visual y auditivo.
  - Los textos alternativos al contenido visual o auditivo benefician a personas ciegas y/o sordas y a aquellos usuarios que deciden anular la descarga de imágenes y/o sonidos, por ejemplo, por motivos de velocidad de acceso a Internet.
  - Los equivalentes no textuales, como pueden ser dibujos o vídeos, benefician a personas analfabetas o con dificultades en la lectura.
- No basarse únicamente en el color. Los textos y gráficos deben comprenderse sin necesidad de ver los colores. El cumplimiento de esta pauta beneficia a personas con dificultades para ver los colores y a usuarios que utilizan pantallas monocromáticas.
- Utilizar marcadores y hojas de estilo apropiadamente. La presentación de los contenidos se debe realizar con hojas de estilo. Con el uso de marcadores de presentación los usuarios que utilizan software especializado tendrán dificultades para entender la estructura de la página.
- Identificar el idioma que se usa. Esta pauta implica usar marcadores que faciliten la pronunciación o interpretación de texto abreviado. Se debe indicar el idioma predominante en cada página y marcar aquellas expresiones que se encuentren en otra lengua. Así, los sintetizadores de voz son capaces de cambiar su pronunciación en función del idioma siempre y cuando se usen los marcadores apropiados.
- Crear tablas que se transformen correctamente. Las tablas sólo se deben utilizar para marcar información tabular (tablas de datos). El uso de tablas con otros fines crea dificultades para los usuarios que usan lectores de pantalla. De la misma forma, las tablas mal estructuradas (por ejemplo, sin encabezados <th>) dificultan la lectura a usuarios que no pueden visualizar la información de forma global: ciegos con lectores de pantalla y/o dispositivos *braille*, deficientes visuales que utilizan magnificadores de pantalla o usuarios con dispositivos de pantalla pequeña.

- Asegurar que las páginas que incorporen nuevas tecnologías se transformen correctamente. Una página basada en tecnologías modernas tiene que ser accesible al desconectarla o al visualizarla con navegadores antiguos. El usuario puede desconectar las tecnologías más modernas para ganar en rapidez de descarga. Sin embargo, los contenidos deben permanecer accesibles.
- Garantizar el control sobre los cambios de contenido sensibles en el tiempo. Es importante pausar o detener el movimiento, el parpadeo, el desplazamiento o la actualización automática de objetos o páginas. Algunas personas con discapacidades cognitivas o visuales no pueden leer el texto en movimiento. Además, los lectores de pantalla no pueden leer texto en movimiento.
- Garantizar el acceso directo a las interfaces de usuario integradas. Se debe asegurar que la interfaz de usuario siga los principios del diseño accesible: acceso independiente del dispositivo a la funcionalidad, la operabilidad del teclado, la voz propia, etc. Cuando un objeto incrustado tiene su “propia interfaz”, ésta debe ser accesible. Si la interfaz del objeto incrustado no puede ser accesible, se debe proporcionar una solución accesible alternativa.
- El diseño debe garantizar la independencia del dispositivo. Usar funciones que permitan la activación de elementos de página a través de una variedad de dispositivos de entrada.
- Usar soluciones provisionales de accesibilidad para que las tecnologías de asistencia y los navegadores antiguos funcionen correctamente.
- Utilizar las tecnologías W3C (según las especificaciones) y seguir las pautas de accesibilidad. Cuando no sea posible utilizar una tecnología W3C, o si al hacerlo el material no se transforma correctamente, proporcionar una versión alternativa del contenido que sea accesible.
- Proporcionar información de contexto y orientación para ayudar a los usuarios a comprender páginas o elementos complejos. Las relaciones complejas entre partes de una página pueden ser difíciles de interpretar para las personas con discapacidades cognitivas y las personas con discapacidades visuales.
- Proporcionar mecanismos de navegación claros y consistentes (información de orientación, barras de navegación, un mapa del sitio, etc.) para aumentar la probabilidad de que una persona encuentre lo que busca en un sitio.
- Asegurar que los documentos sean claros y sencillos para que se puedan entender más fácilmente. Un diseño de página consistente, con gráficos

reconocibles y un lenguaje fácil de entender, benefician en particular a las personas con discapacidades cognitivas o que tienen dificultades para leer.

### 3.1.2.2. Pautas de Accesibilidad al Contenido en la Web 2.0

Con el paso del tiempo, las WCAG 1.0 comenzaron a quedarse obsoletas, ya que debido al avance de las tecnologías web resultaba más difícil verificar sus pautas.

Por ello, aparecen Las Pautas de Accesibilidad al Contenido en la Web 2.0 (Web Content Accessibility Guidelines 2.0, WCAG 2.0 (Chisholm et al., 2008)), que se fundamentan en WCAG 1.0, introduciendo algunos cambios significativos.

A un nivel práctico, algunos de los cambios se muestran igual que las WCAG 1.0. Por ejemplo, los formularios todavía requieren etiquetas, las tablas de datos deben contener cabeceras y las imágenes todavía requieren un texto alternativo, por lo que los desarrolladores web que ya diseñan sitios web accesibles no tendrán que cambiar demasiado sus hábitos.

Por otro lado, las WCAG 2.0 representan un cambio en su filosofía. Los cambios importantes implican que las pautas están centradas en principios más que en técnicas, permitiendo que las pautas sigan siendo relevantes incluso cuando la tecnología cambie. Además, están diseñadas para que su adecuación se pueda verificar de forma fiable.

El cambio de pautas centradas en las técnicas a pautas centradas en principios dio lugar a un número reducido de ideas de nivel superior o principios. Las WCAG 1.0 tenían catorce principios en el nivel superior. En cambio, las WCAG 2.0 presenta únicamente cuatro principios en el nivel superior. Cada uno de estos cuatro principios se indica con una sola palabra:

- PRINCIPIO 1: PERCEPTIBLE. La información y los componentes de la interfaz de usuario deben estar presentables para los usuarios de maneras que puedan percibir.
- PRINCIPIO 2: OPERABLE. Los componentes de la interfaz de usuario y la navegación deben ser operables.
- PRINCIPIO 3: COMPRENSIBLE. La información y el funcionamiento de la interfaz de usuario deben ser comprensibles.
- PRINCIPIO 4: ROBUSTO. El contenido debe ser lo suficientemente robusto como para que pueda ser interpretado de manera confiable por

una amplia variedad de agentes de usuario, incluidas las tecnologías de asistencia.

### 3.1.3. ¿Por qué la Accesibilidad Web es importante?

Hoy en día, usamos internet para muchos aspectos de nuestra vida: educación, empleo, sanidad, entretenimiento... Por ello, es importante que la web sea accesible para poder proporcionar acceso equitativo y en igualdad de oportunidades para todas personas.

El hecho de que una página web sea accesible permite ayudar a que personas con discapacidad participen más activamente en la sociedad, integrándose en ésta en mayor medida, brindándoles así una oportunidad de acceder a la información y de interactuar.

## 3.2. ¿Qué es la Lectura Fácil?

Podemos encontrar dos definiciones ligeramente distintas para este término:

- Adaptación lingüística de un texto que facilita más la lectura de un contenido, aunque no su comprensión.
- Adaptación que hace más fácil tanto la lectura como la comprensión.

La Lectura Fácil no sólo abarca el texto, sino también se refiere a las ilustraciones y maquetación. Se dirige a todas las personas, en especial a aquellas que tienen dificultades lectoras transitorias (inmigración, incorporación tardía a la lectura, escolarización deficiente...) o permanentes (trastornos del aprendizaje, diversidad funcional, senilidad...).

### 3.2.1. Directrices internacionales de la IFLA

En 1997 la IFLA<sup>2</sup>(International Federation of Library Associations and Institutions) creó una serie de Directrices Internacionales que han sido traducidas a todos los idiomas de la Unión Europea, dentro de un proyecto que persigue hacer accesible a todos los ciudadanos los servicios y medios de comunicación.

El objetivo que se pretende con estas directrices es determinar las pautas que debe seguir quien redacta un texto para que su lectura sea fácilmente

---

<sup>2</sup>Para más información, consultar la página <http://www.ifla.org>

comprensible por todos.

Las Directrices de Lectura Fácil van dirigidas a todas las personas que tengan dificultades para leer y entender el idioma del país en el que residen. Por ejemplo, personas con discapacidad cognitiva, colectivos que padecen otro tipo de discapacidad que afecta también a su capacidad de lectura o comprensión, personas con un bajo nivel cultural, personas mayores...

Para hacer un texto que sea comprensible para el mayor número de personas, existen una serie de normas de diseño y de redacción.

#### **3.2.1.1. Normas de diseño**

El diseño de las publicaciones juega un papel importante en la facilidad de lectura. Las recomendaciones más importantes a tener en cuenta en lo que se refiere al diseño son (Moreno y Franco, 2009):

- Elegir letras claras, como Arial, Helvética, Verdana, Times New Roman o Sans Serif, entre otras.
- Utilizar como máximo dos tipos de letras dentro del mismo documento.
- El tamaño de la letra ha de ser grande o configurable si el formato es electrónico.
- No utilizar mayúsculas ni cursivas. Usar la negrita o el subrayado para enfatizar palabras o frases.
- No justificar el texto a la derecha. Lo más apropiado es usar una alineación a la izquierda.
- No poner dibujos como fondo de un texto.
- Intentar utilizar una sola línea para cada oración.
- Evitar separar los elementos constitutivos de la oración, de modo que ésta quede siempre dentro de una misma página.
- No incluir demasiada información en una página.
- No usar nunca la impresión invertida.
- Utilizar colores para los dibujos.
- No usar guiones para separar palabras largas en el margen derecho del texto

### 3.2.1.2. Normas de redacción

El otro tipo de pautas a seguir son las relacionadas con el contenido del texto, que son fundamentales para que todos puedan entender la información que queremos transmitir. Sin embargo, el uso de estas pautas no implica caer en un lenguaje simplista e infantil.

Las pautas relacionadas con el contenido son (Moreno y Franco, 2009):

- Usar un lenguaje sencillo y claro.
- Evitar conceptos abstractos. Si han de usarse, utilizar ejemplos concretos o aclaraciones.
- Emplear vocablos cortos relativos al lenguaje cotidiano hablado.
- Personificar el texto tanto como sea posible. Es mejor decir “usted tiene derecho a...” que “los usuarios del servicio tienen derecho a...”
- Hacer uso de ejemplos prácticos.
- Dirigirse a los lectores de manera respetuosa.
- Incluir una sola idea principal en cada oración.
- Utilizar un lenguaje positivo.
- Emplear preferentemente la voz activa frente a la pasiva.
- No dar por asumidos conocimientos previos sobre el tema en cuestión.
- Ser sistemático al utilizar las palabras.
- No emplear el subjuntivo.
- Tener cuidado con el lenguaje figurativo o metafórico si son vocablos de uso poco común.
- No emplear palabras de otro idioma.
- Mencionar, siempre que sea posible, una dirección de contacto para solicitar más información.
- Evitar el uso de jergas, abreviaturas e iniciales.



### 3.2.2. ¿Por qué la lectura fácil es necesaria?

La lectura fácil es necesaria porque:

- El acceso a la lectura y a la información es un derecho y una necesidad social.
- Leer es un placer que permite compartir ideas, pensamientos y experiencias.
- Muchos textos tienen un exceso de tecnicismos, una sintaxis compleja y una presentación poco clara.
- Existe cierto porcentaje de la población que presenta dificultades lectoras.

## 3.3. Proyectos relacionados

En esta sección comentaremos algunos de los proyectos que existen en la actualidad para afrontar el problema de la accesibilidad web.

### 3.3.1. inSuit

InSuit<sup>3</sup> proporciona accesibilidad y usabilidad web de manera sencilla. Se trata de un producto de apoyo que añade a la página web una capa de información semántica, personalizada por expertos en accesibilidad y usabilidad.

#### 3.3.1.1. ¿Qué nos permite inSuit?

InSuit nos permite:

- Mejorar de manera automática el cumplimiento de muchas de las recomendaciones del W3C en materia de accesibilidad web, haciendo que la web sea más accesible y usable.
- Proporcionar desde la nube las ayudas técnicas para que cada persona pueda navegar de manera adaptada a sus necesidades y preferencias.

#### 3.3.1.2. ¿Qué ofrece inSuit?

Al proporcionar las ayudas desde la nube, *inSuit* está en continua evolución. Cada persona puede navegar de manera adaptada a sus necesidades y preferencias. Ofrece las siguientes ayudas técnicas:

---

<sup>3</sup>Para más información, acceder a la página <https://www.insuit.net/es/>

- inSuit Aloud. La web lee al usuario el contenido y las opciones disponibles, lo que posibilita el acceso a los contenidos y la navegación sin necesidad de ver la pantalla.
- inSuit Voice. El usuario puede hablar a la web para navegar mediante sencillos comandos, sin utilizar las manos.
- inSuit Sound. El usuario puede navegar emitiendo cualquier tipo de sonido, sin necesidad de utilizar las manos ni de vocalizar comandos.
- inSuit Keys. El usuario puede navegar mediante el teclado sin necesidad de utilizar el ratón.
- inSuit Visual. El usuario visualiza textos limpios y puede ajustar el tamaño y la combinación de contraste y color.
- inSuit vKeyboard. El usuario puede navegar mediante el ratón utilizando un teclado virtual y elementos visuales destacados que centran al usuario, simplifican y ayudan a la navegación.
- inSuit ExtAid. El usuario utiliza su propio dispositivo de ayuda a la navegación, con las potentes mejoras de accesibilidad y usabilidad añadidas por la solución.
- inSuit Button. El usuario puede navegar utilizando el teclado como un dispositivo pulsador que acciona un teclado de barrido. El usuario interactúa con el interfaz con punteros de boca, diademas con puntero, con los pies, pulsadores externos, etc.

#### 3.3.1.3. Limitaciones de inSuit

A pesar tratarse de una aplicación que cubre una gran cantidad de servicios y funcionalidades, es una herramienta de pago. Por ello, debido a su precio, no está disponible para todo tipo de usuarios. Además, requiere disponer de una página web para poder implantar los servicios que ofrece.

#### 3.3.2. NavegaFácil

Como podemos ver en la figura 3.1, NavegaFácil (Bautista et al., 2018) es una aplicación desarrollada para un trabajo de fin de grado por dos alumnos de la Universidad Complutense de Madrid y que ha servido como inspiración a la hora de la realización de la extensión para este proyecto.

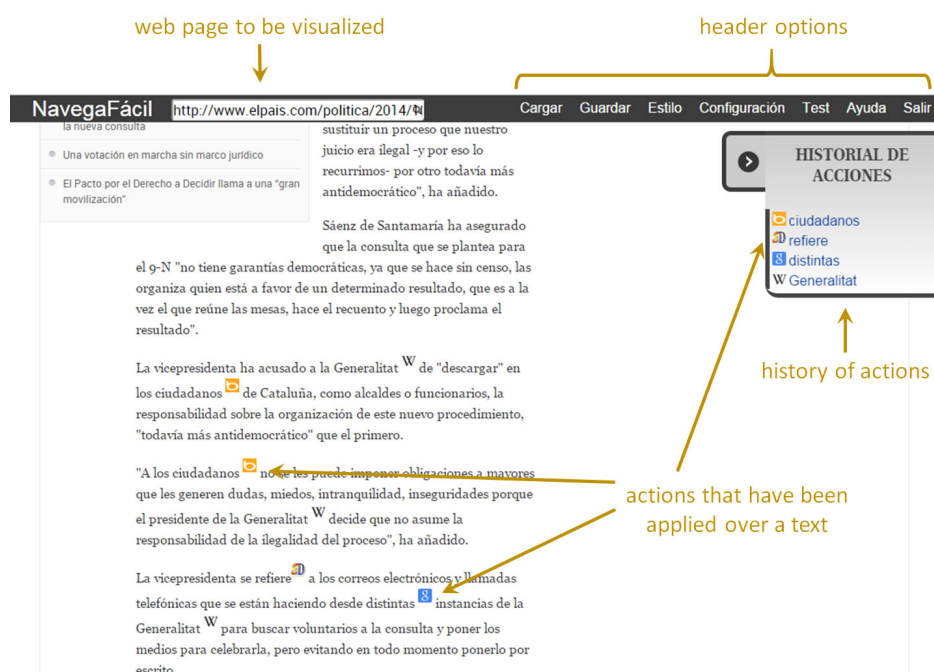


Figura 3.1: Diseño de NavegaFácil

### 3.3.2.1. ¿Qué nos permite NavegaFácil?

NavegaFácil es una web que permite a los usuarios reproducir el contenido original de cualquier otra web añadiéndole diferentes funcionalidades con el objetivo de facilitar la comprensión de toda la información escrita. NavegaFácil permitirá a los usuarios enriquecer el contenido original de cualquier otra web añadiendo diversas herramientas y funcionalidades con el objetivo de facilitar su comprensión.

### 3.3.2.2. ¿Qué ofrece NavegaFácil?

Las herramientas de la web buscan hacer lo más universal posible el concepto de lectura fácil. Por ello se componen de los servicios web y de diversas funcionalidades con el fin de adaptar cualquier texto de la web al mayor colectivo de personas posibles, ya sea por el tipo de letra, el color del fondo... Algunas de las herramientas que ofrece son:

- Definición. Obtención de la definición de una palabra seleccionada por el usuario en el contenido de la web.
- Sinónimos y antónimos. Obtención de los sinónimos y antónimos de

una palabra seleccionada por el usuario en el contenido de la web.

- Sinónimos (origen latinoamericano). Obtención de los sinónimos (de origen latinoamericano) de una palabra seleccionada por el usuario en el contenido de la web.
- Lematización. Obtención de la lematización de una palabra seleccionada por el usuario en el contenido de la web cuyo resultado es el lema (singular para sustantivos, masculino singular para adjetivos, infinitivo para verbos).
- Imágenes. Obtención de imágenes relacionadas con una palabra o texto seleccionado por el usuario en el contenido de la web.
- Traductor. Obtención de la traducción del texto seleccionado por el usuario en el contenido de la web.
- Lectura. Obtención de la reproducción sonora del texto seleccionado por el usuario en el contenido de la web.
- Wikipedia. Obtención del resultado de Wikipedia sobre la palabra seleccionada por el usuario en el contenido de la web.
- Difíciles. Obtención del contenido de la web resaltando las palabras difíciles mediante el servicio web “Difíciles”.
- Filtrado del contenido. Obtención del contenido de la web cambiando las palabras malsonantes por otras más apropiadas y filtrado de las búsquedas e imágenes de contenido inapropiado, permitiendo a los usuarios generar un contenido apto para un mayor rango de posibles receptores de su trabajo.
- Guardar y cargar. Permite guardar y recuperar todo el contenido que el usuario haya necesitado buscar en la web.

### 3.3.2.3. Limitaciones de NavegaFácil

Una de las principales limitaciones que presenta NavegaFácil era la necesidad de parsear toda una página web, siendo un proceso pesado. Además, presenta grandes problemas de seguridad. No está preparado para que se puedan integrar nuevas funcionalidades fácilmente.

### 3.3.3. ATbar

ATbar<sup>4</sup> es un plugin que está disponible para los navegadores más populares. Como vemos en la figura 3.2, se trata de una barra de herramientas

---

<sup>4</sup>Para más información visitar la página de ATbar <https://www.atbar.org/>

para ayudar a los usuarios a personalizar la forma en que ven e interactúan con las páginas web.



Figura 3.2: Diseño de la barra de herramientas

### 3.3.3.1. ¿Qué nos permite ATbar?

ATbar es gratuito y permite cambiar el aspecto de las páginas web, aumentar y disminuir el tamaño de las fuentes, hacer que el texto se lea en voz alta, utilizar superposiciones de colores...

ATbar puede aumentar la accesibilidad de un sitio web y ofrecer una experiencia de usuario personalizada. Está diseñado para quienes no tengan sus tecnologías de asistencia a mano y necesiten una forma rápida de acceder al texto en la pantalla con aumento y/o texto por voz, etc.

Puede ayudar a las personas con baja visión, dislexia y otras dificultades de lectura, así como a todos aquellos que deseen simplemente reducir el resplandor del texto negro sobre fondos blancos brillantes. No está diseñado para usuarios habituales de lectores de pantalla que necesitan su tecnología de asistencia para acceder a la computadora y a su navegador.

### 3.3.3.2. ¿Qué ofrece ATbar?

Como hemos visto en la figura 3.2, explicamos las herramientas que ofrece ATbar de izquierda a derecha:

- Reparar la web. Sirve para ayudar a solucionar los problemas de accesibilidad web que implican que muchas personas con discapacidad no puedan usar partes de Internet. Todavía está en desarrollo.
- Maximizar texto. Permite aumentar el tamaño del texto. Si el cambio de tamaño del texto hace que el sitio no se pueda leer, es posible que el usuario deba utilizar la función de zoom del navegador para aumentar tanto la imagen como el tamaño del texto.
- Minimizar texto. Permite disminuir el tamaño del texto. Si el cambio de tamaño del texto hace que el sitio no se pueda leer, es posible que el

usuario deba utilizar la función de zoom del navegador para aumentar tanto la imagen como el tamaño del texto.

- Cambios de fuente. Permite cambiar la fuente y el espaciado entre líneas.
- Corrector ortográfico. Si este botón está activo mientras el usuario escribe en un formulario de texto sin formato, las sugerencias se muestran debajo del área de texto después de una pausa. Puede no funcionar con los editores de texto enriquecido que ya tienen correctores ortográficos.
- Diccionario. Si el usuario resalta una palabra en la página web y pulsa este botón, aparecerá una definición de Wiktionary<sup>5</sup>.
- Texto por voz. El usuario debe seleccionar una parte del texto y pulsar este botón para que solo se lea ese texto.
- Predicción textual. Tras pulsar este botón, muestra a los usuarios que estén redactando en un formulario el texto predictivo de la palabra que escriben. Funcionará con la mayoría de los formularios de edición de texto sin formato. Puede ayudar con la ortografía, la búsqueda de palabras e incluso puede acelerar la escritura. Hay algunos editores de texto enriquecido y formularios basados en Flash<sup>6</sup> donde el cuadro de predicción puede no aparecer o verse afectado por el diseño de la página web.
- Cambiar estilos. Permite cambiar la apariencia de la barra y la página. Algunos cambios de color en la página web también tienen un impacto en la barra de herramientas.
- Cambios de color de fondo de la página. Proporciona un matiz de color en toda la página web. Si el usuario pulsa este botón, aparecerá una selección de colores: crema, rosa, azul pálido y verde pálido. Estos botones activan y desactivan el color de fondo de la página.
- Ayuda. Este botón abre una ventana nueva en esta dirección: [http://en.wiki.atbar.org/wiki/Main\\_Page](http://en.wiki.atbar.org/wiki/Main_Page).
- Revertir. Permite restaurar la página web a su estilo original.
- Cerrar. Cierra la barra de herramientas de Atbar.

---

<sup>5</sup>Es un proyecto de diccionario libre de la Fundación Wikimedia, que contiene definiciones, traducciones, etimologías, sinónimos y pronunciaciones de palabras en múltiples idiomas.

<sup>6</sup>Flash es una tecnología para crear animaciones gráficas vectoriales independientes del navegador y que necesitan poco ancho de banda para mostrarse en los sitios web.

### 3.3.3.3. Limitaciones de ATbar

Una vez estudiada la aplicación, encontramos las siguientes limitaciones:

- Para poder abrir la barra de herramientas, es necesario clicar en el icono del plugin de la aplicación, limitando mucho el número de usuarios que lo usen.
- Una vez la barra de herramientas está abierta, los iconos que presenta no resultan intuitivos, a pesar de que muestre pequeñas descripciones al pasar el cursor en los botones. Para saber cómo funcionan es necesario leerse la guía del plugin<sup>7</sup>.
- Si cambias la apariencia de la página, modificando sus estilos, muchas de las opciones que ofrece no funcionan.

### 3.3.4. Extensión WAVE Evaluation Tool

WAVE<sup>8</sup> es una herramienta de evaluación de accesibilidad web desarrollada por *WebAIM.org*. Proporciona comentarios visuales sobre la accesibilidad del contenido web mediante la inyección de iconos e indicadores en su página.

WAVE facilita la evaluación humana y educa sobre los problemas de accesibilidad. Todo el análisis se realiza dentro del navegador Chrome, lo que permite una valoración segura.

En la figura 3.3 mostramos un ejemplo del análisis que ofrece.

### 3.3.5. OpenDyslexic y OpenDyslexic Font-Helperbird-Free

Como vemos en la figura 3.4, los plugins OpenDyslexic<sup>9</sup> y OpenDyslexic Font-Helperbird-Free<sup>10</sup> ofrecen la posibilidad de aumentar la legibilidad para los lectores con dislexia. Esta extensión anula todas las fuentes en las páginas web con la fuente OpenDyslexic y formatea las páginas para que sean más fáciles de leer. En la figura 3.5 vemos el resultado tras aplicarlos.

---

<sup>7</sup>Podemos ver la guía en el siguiente enlace: <http://en.wiki.atbar.org/wiki/Category:Guides>

<sup>8</sup>Lo podemos encontrar en <https://chrome.google.com/webstore/detail/wave-evaluation-tool/jbbplnpkjmeebjpijfedlgcdilcofh>

<sup>9</sup>Lo podemos encontrar en <https://chrome.google.com/webstore/detail/opendyslexic/cdnagpfjopgaggbmfgbiinmmdbdcglnam>

<sup>10</sup>Lo podemos encontrar en <https://chrome.google.com/webstore/detail/opendyslexic-font-helperb/ahmapmilbkfamljbpghfndeemhnaime>

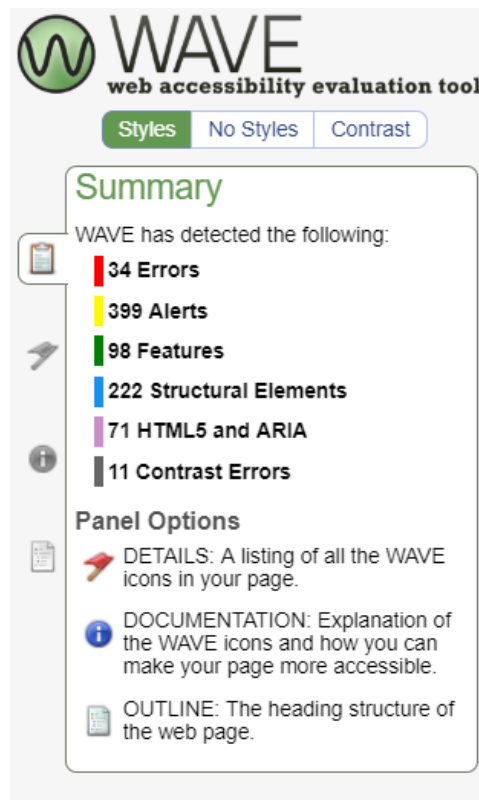
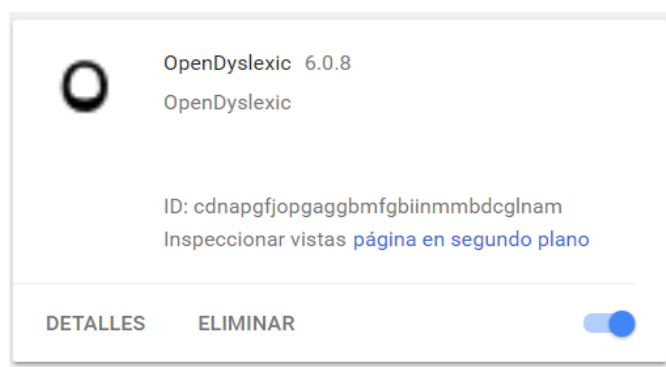
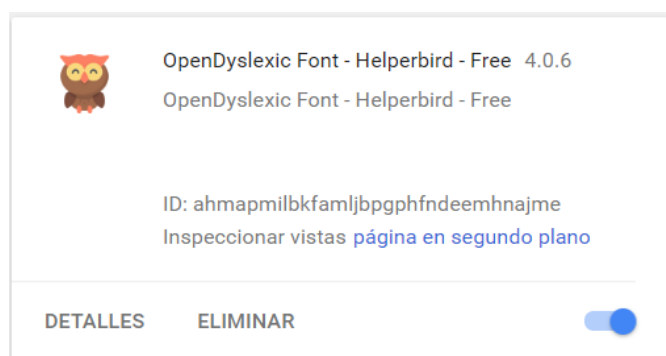


Figura 3.3: Resultados tras ejecutar WAVE





(a) Plugin OpenDyslexic



(b) Plugin OpenDyslexic Font-Helperbird-Free

Figura 3.4: Plugins OpenDyslexic y OpenDyslexic Font-Helperbird-Free

Tocadas las 17.30, el profesor Giuseppe Conte, un absoluto desconocido para los italianos, llegó solo en un taxi a la colina del Quirinal desde su bufete de abogados en el centro de Roma. Pagó la carrera en metálico y entró en el estudio de cristalerías del Palacio presidencial por primera vez en su vida. Una hora y 45 minutos después, una eternidad comparado con el resto de consultas, volvió a salir con la misma carpeta blanca que traía bajo el brazo y el mandato del presidente Mattarella de formar Gobierno. El profesor, 54 años, separado y padre de un hijo de 10 años, será el presidente del Consejo de Ministros de Italia número 65 en los últimos 72 años. Sin embargo, él prefirió hablar de sí mismo como el “abogado defensor de todos los italianos”.

(a) Antes de aplicar los cambios de los plugins

Tocadas las 17.30, el profesor Giuseppe Conte, un absoluto desconocido para los italianos, llegó solo en un taxi a la colina del Quirinal desde su bufete de abogados en el centro de Roma. Pagó la carrera en metálico y entró en el estudio de cristalerías del Palacio presidencial por primera vez en su vida. Una hora y 45 minutos después, una eternidad comparado con el resto de consultas, volvió a salir con la misma carpeta blanca que traía bajo el brazo y el mandato del presidente Mattarella de formar Gobierno. El profesor, 54 años, separado y padre de un hijo de 10 años, será el presidente del Consejo de Ministros de Italia número 65 en los últimos 72 años. Sin embargo, él prefirió hablar de sí mismo como el “abogado defensor de todos los italianos”.

(b) Después de aplicar los cambios de los plugins

Figura 3.5: Aplicando los plugins OpenDyslexic

## Capítulo 4

# Tecnología

En este capítulo hablaremos sobre la tecnología escogida para desarrollar el proyecto, una extensión para Google Chrome. Describiremos los distintos motivos para seleccionarla frente a otras alternativas.

Trazaremos finas pinceladas acerca de los orígenes de dicho navegador, las distintas versiones que existen en la actualidad y sus principales características.

Describiremos algunos tipos de extensiones que nos podemos encontrar en internet, así como las más usadas según las necesidades de los usuarios. Explicaremos qué es una extensión web, definiendo su estructura y arquitectura generales.

### 4.1. Orígenes de Google Chrome

Chrome (Leonardo, 2012) fue presentado por primera vez de manera oficial el 2 de Septiembre de 2008 para Microsoft Windows (únicamente para XP y versiones posteriores) en 43 idiomas siendo una versión beta. Obtuvo en un breve periodo de tiempo el 1 % del mercado de navegadores. Un tiempo después, el 11 de Diciembre de 2008 fue lanzado de manera oficial.

Esta primera versión pasó las pruebas de Acid1<sup>1</sup> y Acid2<sup>2</sup> (esta última con un pequeño error). Obtuvo 79 puntos de 100 en la prueba de Acid3<sup>3</sup>,

---

<sup>1</sup>Originalmente llamado Box Acid Test, es una página de prueba para los navegadores web. Se desarrolló en octubre de 1998 y fue importante para establecer una base de referencia para la interoperabilidad de los primeros navegadores web.

<sup>2</sup>Es una página de prueba publicada para detectar fallos de renderización. Fue desarrollada en el espíritu de Acid1. Fue lanzada el 13 de abril de 2005.

<sup>3</sup>Pone a prueba los navegadores con los estándares web. Desarrollo desde abril de 2007 y lanzado el 3 de marzo de 2008.

siendo superior a Internet Explorer y Firefox, pero inferior a Opera.

El 9 de diciembre de 2009, Google anunció la publicación de las versiones beta de Chrome para Mac OS X y Linux. A principios de 2010 ya cuenta con más de 1500 funciones disponibles. En marzo, surgen los controles de privacidad y el traductor de Google. El 25 de mayo de 2010 Google anunció la versión estable de su navegador Google Chrome versión 5 simultáneamente para todas las plataformas Microsoft Windows, Mac OS X y Linux. Al mes siguiente, se integra Flash a Chrome. En febrero de 2012 se lanza la versión beta para Android. En junio, surgen las versiones de Chrome para iPhone y iPad.

Actualmente, el navegador está disponible para la plataforma Microsoft Windows en más de 50 idiomas. La versión para sistemas Mac OS X y Linux se encuentra actualmente en desarrollo, con disponibilidades de versiones beta en ambos sistemas operativos.

#### 4.1.1. Versiones

Google Chrome es el navegador web más utilizado en todo el mundo. Actualmente, existen cinco versiones (Velasco, 2017) de Google Chrome distintas, cada una de ellas pensadas para una finalidad en concreto.



Figura 4.1: Versiones de Chrome ordenadas de menor a mayor estabilidad

##### GOOGLE CHROME STABLE

Se trata de la más estable de Chrome y la más usada por los usuarios. Es la versión final, libre de errores lista para funcionar, proporcionando la mejor experiencia, seguridad y rendimiento mientras navegamos en la web. Recibe las actualizaciones más importantes, siendo probadas antes por las demás versiones de desarrollo.

### **GOOGLE CHROME BETA**

Es considerada como la versión más equilibrada que ofrece Google para dispositivos Android. Funciona igual que Google Chrome Stable y suele estar adelantada, ya que Google suele probar aquí todas las funciones que, con el tiempo, llegarán a la versión final, por lo que existe la posibilidad de encontrar algunos errores.

Chrome Beta es recomendada para aquellos usuarios que sufran problemas con la versión estable o para los que quieran disfrutar antes de las novedades. Esta versión se suele actualizar todas las semanas y suele recibir una actualización mayor cada seis semanas.

### **GOOGLE CHROME DEV**

Es inferior de la versión Beta y Stable, por lo que es más inestable. Se trata de una variante pensada para desarrolladores, para detectar los errores, solucionarlos y desarrollar nuevas versiones. Su principal objetivo es que los administradores sean capaces de comprobar que sus páginas web funcionen bien con las nuevas extensiones y complementos instalados.

### **GOOGLE CHROME CANARY**

Es la versión más inestable de todas. Se genera automáticamente desde los servidores de Google con todos los cambios que se han realizado en el código sin ningún tipo de comprobación. Se caracteriza por continuas actualizaciones, errores y su función experimental. Esta variante está destinada a los desarrolladores, para poder comunicar los errores existentes. Puede recibir hasta siete actualizaciones a la semana.

### **GOOGLE CHROMIUM**

Chromium es uno de los proyectos más interesantes de la compañía gracias a su diversidad. Su objetivo principal es proporcionar un navegador con mayor estabilidad, velocidad y seguridad además de incluir una interfaz de usuario sencilla y eficiente. Chromium es el navegador base del que está construido Chrome y tiene sus mismas características de diseño, pero con un logotipo ligeramente diferente y sin el apoyo comercial y técnico de la compañía Google.

## **4.2. ¿Qué es una extensión?**

Las extensiones son pequeños programas que se instalan dentro del navegador (Google Chrome, Firefox, Opera, etc.) que permiten modificar y mejorar la funcionalidad del navegador, facilitando la experiencia de los usuarios una vez que son instaladas. Suelen tener un tamaño bastante pequeño (no más de 1MB). Se desarrollan utilizando tecnologías web tales como HTML,

Javascript y CSS.

Cada usuario puede seleccionar la extensión que quiera en función de sus necesidades e instalarla en su navegador. Hoy en día existe gran variedad de extensiones para cubrir las necesidades de los usuarios. Pueden ser gratis o de pago, pueden ser creadas por los desarrolladores de Google o por terceros.

#### 4.2.1. ¿Por qué una extensión para Chrome?

Uno de los principales motivos para desarrollar una extensión para Google Chrome es que es uno de los navegadores más utilizados a nivel mundial por los usuarios.

En la figura 4.2 podemos comprobar en el diagrama de sectores<sup>4</sup> cómo el porcentaje de uso de los usuarios de dicho navegador es superior en casi un 40 % respecto del resto.

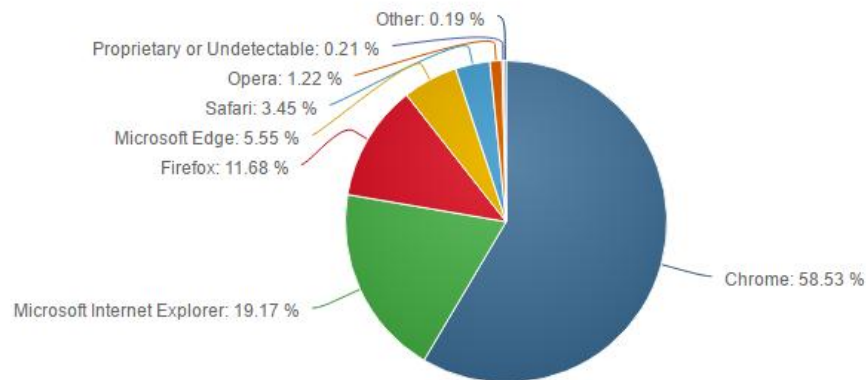


Figura 4.2: Uso de distintos tipos navegadores

Escogiendo Chrome como navegador para el desarrollo del proyecto proporcionaríamos soporte a un amplio número de usuarios que usan internet a diario. Sin embargo, la navegación no es igual para todos. Cada usuario puede presentar distintos problemas a la hora de comprender un artículo, por ejemplo, que contenga palabras de las que desconoce su significado. Desarrollando una extensión, podremos cubrir esta y otros tipos de problemática para un sector concreto de usuarios en todas las páginas a las que desee acceder.

<sup>4</sup>Diagrama extraído del artículo <https://www.muycomputer.com/2017/03/01/navegadores-web-febrero/>

Otro motivo que impulsó al desarrollo de una extensión es aportar la capacidad de mejorar los servicios que nos proporciona el navegador para aquellas personas con mayor dificultad.

Una característica importante a tener en cuenta, es que es muy sencillo habilitar un extensión de Chrome, como explicaremos en otra sección.

#### 4.2.2. Características básicas de Google Chrome

Google Chrome proporciona una serie de características que lo hacen diferente y completo. Presenta una interfaz simple y clara, cuenta con gran estabilidad y seguridad. La mejor característica que proporciona para los usuarios es la rapidez. Indicamos a continuación algunas características básicas (DKEN2302, 2015) (Chauvin, 2012):

- NAVEGACIÓN SEGURA: Para navegar de manera más segura, Google Chrome advierte al usuario cuando está a punto de visitar un sitio sospechoso o inseguro. Al realizar la navegación mediante pestañas independientes, que están incomunicadas, se impide el robo de la información.
- EFICACIA: Google Chrome está diseñado para soportar aplicaciones web complejas y es compatible con los lenguajes de programación actuales.
- SOPORTA MEJORAS Y ACTUALIZACIONES: Está disponible para Windows, Mac y Linux, al igual que sus continuas mejoras, cuya finalidad es que el navegador sea más rápido, estable y funcional.
- VELOCIDAD: El objetivo principal de Chrome es la velocidad de navegación, desde su ejecución hasta la carga de aplicaciones web complejas.
- PRODUCTIVIDAD: El usuario puede acceder a sus marcadores, pestañas abiertas e historial desde cualquier dispositivo que sea compatible con Chrome.
- COMPATIBILIDAD: Google Chrome siempre se encuentra en constante crecimiento por parte de sus desarrolladores, lo que permite incluir nuevas extensiones para mejorar la compatibilidad con otras aplicaciones.
- INTERFAZ SENCILLA Y FUNCIONAL: Está diseñado para ser lo más sencillo posible.
- ADMINISTRADOR DE TAREAS: Chrome contiene un administrador de tareas que nos indica qué recursos se están consumiendo y en qué páginas.

- PRIVACIDAD: A través del modo de incógnito, Chrome permite controlar la información de privacidad y protegerla, navegando en la web sin guardar el historial de navegación ni las búsquedas, permitiendo así al usuario navegar de manera anónima.
- MARCADORES INSTANTÁNEOS: Si el usuario reconoce un sitio de interés, puede guardarlo como sitio preferido en los Marcadores de Google Chrome. Además, el navegador también permite generar carpetas para organizar nuestros marcadores.

### 4.3. Estructura general de las extensiones en Google Chrome

Como ya hemos mencionado, las extensiones son pequeños programas software que pueden modificar y mejorar la funcionalidad del navegador Chrome. Por ello, es importante conocer su estructura general<sup>5</sup>.

Para desarrollar una extensión, lo primero que hay que hacer es crear un fichero llamado *manifest.json*. Se trata de un archivo que contiene toda la configuración, propiedades e información sobre la extensión. Se compone de datos formateados en JSON, que describen metadatos de la extensión, los permisos de los que dispone y los otros componentes de la extensión. En la figura 4.3 podemos ver cuáles son los elementos de los que se compone la estructura de la extensión y cuya configuración deberá estar reflejada en el fichero de manifiesto.

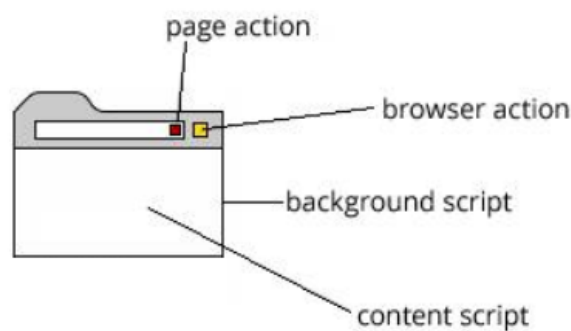


Figura 4.3: Estructura general de una extensión para Chrome

<sup>5</sup>Más información en <https://developer.chrome.com/extensions/overview>



A continuación, explicaremos brevemente qué es cada elemento del que se compone esta estructura:

- **CONTENT SCRIPT** (*Script de contenido*): se utilizan para tener acceso al DOM<sup>6</sup> de la página. Son ficheros CSS y JavaScript que se inyectan en las páginas por las que los usuarios navegan. Se ejecutan después de que la página web en la que el usuario se encuentre se haya cargado.
- **BACKGROUND SCRIPT** (*Script de fondo*): son los controladores de nuestra aplicación. Se ejecuta cuando se inicia Chrome y siempre está escuchando todos los posibles eventos que genere la extensión en chrome. Cuánto más compleja sea la extensión, más necesario será el script de fondo.
- **BROWSER ACTION** (*Acción del navegador*): es la parte más simple de la interfaz de usuario para extensiones. Una acción del navegador es un botón (el botón amarillo de la figura 4.3 ) que se agrega a la barra de herramientas principal a la derecha del omnibox<sup>7</sup>. La extensión puede cambiar el icono y mostrar popups emergentes, que se crean utilizando HTML y se dimensionan de forma dinámica en función de los contenidos.
- **PAGE ACTION** (*Acción de página*): Son similares a las acciones del navegador, pero están en el omnibox y solo aparecen en determinadas páginas (el botón rojo de la figura 4.3 ).
- Además, podemos añadir todos los archivos necesarios para desarrollar nuestra aplicación, como imágenes, bibliotecas de JavaScript...

Todos los archivos deben estar contenidos en una sola carpeta. De esta manera, cuando se distribuye la extensión, el contenido de la carpeta se empaqueta en un archivo ZIP con sufijo .crx, como explicaremos en otro capítulo más detalladamente.

Además, se distinguen dos tipos de extensiones que podemos construir:

- **ACCIONES DE PÁGINA**: acciones que dependen de la página.
- **ACCIONES DE NAVEGADOR**: acciones que no son dependientes de la página.

En la figura 4.4 mostramos un ejemplo de cada uno de los distintos tipos. Como podemos ver, el *Page Action* aparece únicamente cuando realizamos una búsqueda en una página determinada. Por otro lado, el *Browser Action*, en este caso la extensión *AdBlock*, se encuentra presente en todas las páginas.

---

<sup>6</sup>Document Object Model (modelo de objeto de documento)

<sup>7</sup>La barra de direcciones combinada con el cuadro de búsqueda de Google.

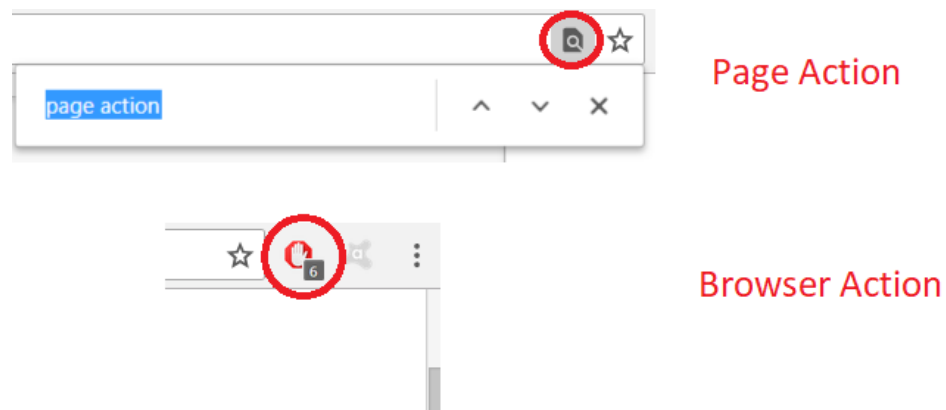


Figura 4.4: Page Action vs Browser Action

Es importante distinguir cuándo usar cada tipo según la finalidad de la extensión. Debemos utilizar las acciones del navegador cuando la funcionalidad de la extensión tiene sentido en la mayoría de las páginas. En caso contrario, es conveniente usar las acciones de página.

Independientemente del tipo de extensión que desarrollemos, Chrome cuenta con una API (un conjunto de subrutinas, funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción) que nos permite añadir diversas funcionalidades para adaptarla a las necesidades del usuario, como por ejemplo:

- Inyectar código en las páginas que visita el usuario.
- Obtener datos de navegación (URL...), marcadores, historial...
- Abrir ventanas o pestañas.
- Sobreescribir la ventana de historial o marcadores.
- Administrar las otras extensiones.
- Sincronizar datos entre instalaciones.

#### 4.3.1. Arquitectura de la ventana emergente

Las extensiones pueden contener ventanas emergentes, es decir, páginas HTML comunes que muestran la interfaz de usuario de la extensión. Por ejemplo, una acción del navegador (el *botón* amarillo de la figura 4.5) puede tener una ventana emergente (el cuadro verde donde pone “popup” de la figura 4.5), que se implementa mediante un archivo HTML (“popup.html”).

Cualquier extensión puede tener una página de opciones que permite a los usuarios personalizar cómo funciona la extensión.

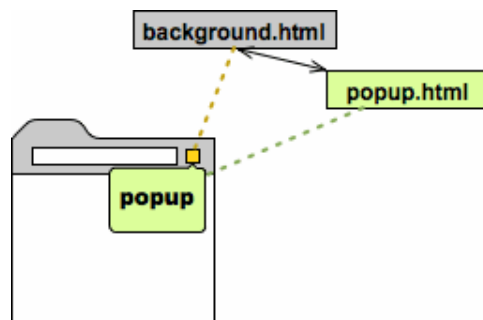


Figura 4.5: Arquitectura de la ventana emergente

Las páginas HTML dentro de una extensión tienen acceso completo a los DOM respectivos y pueden invocar funciones entre sí.

La figura 4.5 muestra la arquitectura de la ventana emergente de una acción del navegador. Los contenidos de la ventana emergente son una página web definida por un archivo HTML (“popup.html”). Esta extensión también tiene una página de fondo (“background.html”). La página de fondo puede estar definida por la extensión .html (en caso de querer incluir distintos archivos JavaScript) o .js (en caso de incluir un único archivo JavaScript). La ventana emergente no necesita duplicar el código que está en la página de fondo porque la ventana emergente puede invocar funciones en la página de fondo.

Si la extensión necesita interactuar con páginas web, es necesario realizar uno o más scripts de contenido, dependiendo de la dificultad de la extensión a desarrollar. Como ya mencionamos, un script de contenido es un JavaScript que se ejecuta en el contexto de una página que se ha cargado en el navegador. Los scripts de contenido pueden leer detalles de las páginas web que visita el navegador y pueden realizar cambios en las páginas. El script de contenido puede leer y modificar el DOM para la página web mostrada. Sin embargo, no puede modificar el DOM de la página de fondo de su extensión principal.

## 4.4. Mundos aislados en las extensiones de Google Chrome

Una de las características más potentes de las extensiones de Google Chrome es la capacidad de comunicarse con páginas web, donde los scripts

de contenido interactúan con el DOM<sup>8</sup>. El DOM, como mostramos en la figura 4.6 es una escritura de datos similar a un árbol, que representa el HTML de una página web.

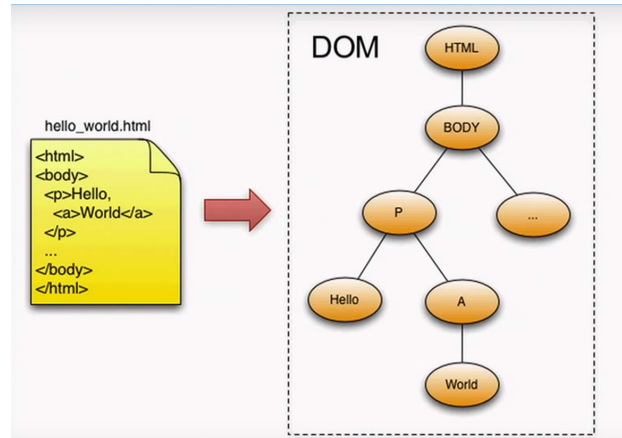


Figura 4.6: Ejemplo de una estructura DOM

Con las extensiones, el DOM de una página web puede ser compartido entre las páginas Javascript y cada uno de los scripts de contenido, como se muestra en la figura 4.7.

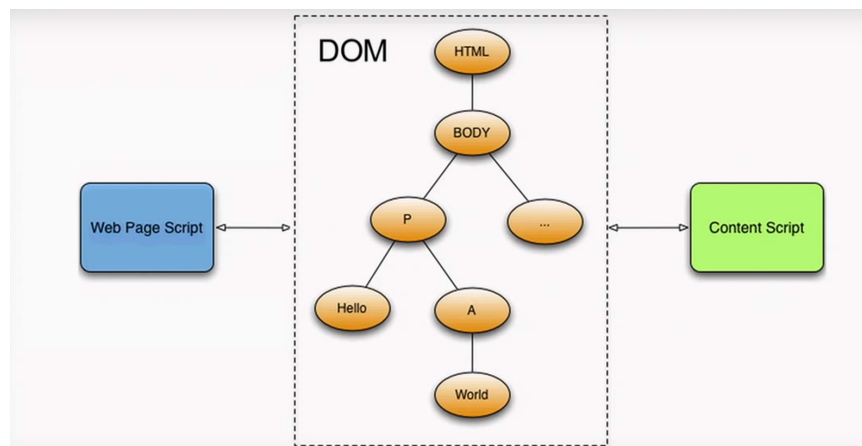


Figura 4.7: DOM compartido

Cualquiera de los scripts puede modificar el DOM. Para compartir la estructura DOM entre páginas web y scripts de contenido es importante que estén aislados entre sí, por razones de seguridad o para evitar que las

<sup>8</sup>Document Object Model (modelo de objeto de documento)

bibliotecas de cada fichero entren en conflicto, entre otros problemas. Como mostramos en la figura 4.8, cada script que está accediendo al DOM se ejecuta en su propio mundo aislado.

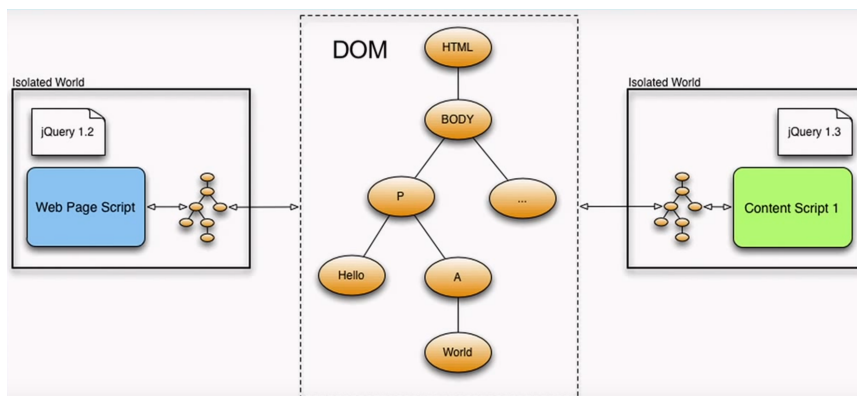


Figura 4.8: DOM compartido con mundos aislados

Cuando realizamos una extensión en Google Chrome, podemos destacar dos mundos aislados<sup>9</sup>. Un mundo se correspondería con el DOM de la página web en la que nos encontramos y otro mundo sería la extensión que desarrollamos. Para que el paso de mensajes se pueda llevar a cabo es necesario que ambos mundos sean capaces de comunicarse.

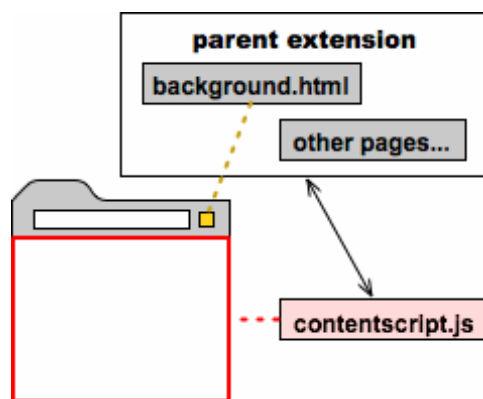


Figura 4.9: Arquitectura que muestra el intercambio de información

Como mostramos en la figura 4.9, el paso de mensajes permite que las diferentes partes de la extensión se comuniquen. Los scripts de contenido se ejecutan en un entorno de JavaScript separado del resto de la extensión para

<sup>9</sup>Más información en [https://developer.chrome.com/extensions/content\\_scripts](https://developer.chrome.com/extensions/content_scripts)

permitirles el acceso directo al DOM de una página web. Ambos mundos se comunican a través de los ficheros de contenido y la página de fondo de la extensión. Existen dos tipos de páginas: de fondo persistentes y páginas de eventos. La primera está activa todo el tiempo. La segunda solo está activa cuando es necesario.

## Capítulo 5

# Diseño centrado en el usuario

El principal objetivo del proyecto es que esta extensión llegue al mayor número de usuarios que presentan dificultades a la hora de comprender los distintos textos que ofrece cualquier página web.

Como desarrolladores web, cuando creamos cualquier aplicación, es común pensar que será útil y fácil de usar. Tendemos a creer que los elementos de la aplicación están bien organizados y colocados en lugares visibles, diseñados de manera clara y bien estructurados, de tal manera que generaría una experiencia agradable y totalmente funcional para todo usuario a la que esté destinada.

Sin embargo, muchas veces no estamos en lo cierto. Quizá no se trate de una herramienta fácil de usar o que se haya implementado funcionalidad que realmente no es útil de cara a los usuarios finales, obteniendo una aplicación inservible.

Para evitar estos problemas, decidimos desarrollar la aplicación siguiendo un diseño centrado en el usuario desde la captura de requisitos hasta la evaluación final del sistema.

Un diseño centrado en el usuario se basa en observar y analizar cómo un grupo de usuarios reales utiliza la aplicación teniendo en cuenta los problemas que se encuentran para poder solucionarlos posteriormente.

Por tanto, su objetivo principal es encontrar todos los posibles problemas de usabilidad que generen mayor impacto al usuario para poder resolverlos de la manera más eficaz y rápida. Es importante valorar las necesidades de los usuarios finales y la manera en la que interactúan con la aplicación para encontrar los problemas a los que se afrontan con la mayor brevedad. Para ello, se puede trabajar con los usuarios finales en diferentes momentos:

- Antes de empezar a crear la aplicación, para detectar rápidamente los errores de diseño.
- Durante el desarrollo de la aplicación, para no desviarnos y centrarnos en nuestros objetivos para conseguir la funcionalidad y usabilidad deseadas.
- Una vez desarrollada y lanzada, para conocer los pasos de deberemos seguir para mejorar en una siguiente versión.

Por tanto, gracias a esta aproximación, aseguramos que nuestra aplicación sea realmente útil, presente un alto nivel de usabilidad y funcionalidad para nuestros usuarios finales.

### 5.1. Colegio Estudio 3 AFANIAS

El Colegio de Educación Especial Estudio 3 AFANIAS<sup>1</sup> es un centro concertado con La Consejería de Educación, Juventud y Deporte de la comunidad de Madrid. En el colegio Estudio 3 AFANIAS se escolariza alumnos entre los tres y los veintiún años con necesidades educativas especiales asociadas a discapacidad intelectual, plurideficiencias y trastorno generalizado del desarrollo, en las etapas de Educación Infantil, Educación Básica Obligatoria y Programas para la Transición a la Vida Adulta.

Gracias a Raquel Hervás y Susana Bautista, tutoras del proyecto, conseguimos concertar una reunión a principios de diciembre con el personal del colegio Estudio 3 AFANIAS para presentar una primera aproximación de la aplicación que íbamos a desarrollar.

En nuestra primera visita al centro nos reunimos con Carlos Roldán (responsable TIC del colegio) que nos ayudaría a comprobar la utilidad y usabilidad de nuestra extensión, nos hizo una pequeña visita guiada por el colegio, presentándonos a los alumnos que participarían en el test de evaluación.

### 5.2. Colegio Estudio 3 AFANIAS: Primera reunión

Una vez reunidos en el colegio con el responsable TIC del centro, expusimos que nuestra intención era desarrollar una extensión en Google Chrome para ayudar a sus alumnos (u otras personas que lo necesitaran) a navegar mejor en la red.

---

<sup>1</sup>Podemos encontrar más información en: <http://colegioestudio3.blogspot.com/>



Nuestro objetivo principal era conseguir información acerca de las necesidades básicas de los alumnos. Así, podríamos focalizar mejor el desarrollo de nuestra aplicación en mejorar su navegación en internet, evitando la posibilidad de desarrollar una herramienta que no cubriera las necesidades de los usuarios a los que está enfocada la aplicación.

Nuestras propuestas pretendían resolver cuestiones relacionadas con la interfaz y funcionalidad de la aplicación. Por ello, nos planteamos las cuestiones siguientes:

- Cuál era la manera más sencilla para obtener palabras que plantearan dificultades para los usuarios: seleccionando el texto de la página, escribiéndolo...
- Cómo debíamos presentar la interfaz de nuestra herramienta: mostrando una barra de herramientas en todas las páginas, mostrar un menú tras seleccionar la palabra que les presenta dificultades...
- Cuáles serían las herramientas más útiles para facilitar la lectura de los usuarios: sinónimos, antónimos, definiciones...
- Qué opciones podrían resultar interesantes: guardar y exportar los datos que ellos buscaran, mostrar marcas de guardado en el documento...

Ante esta problemática de no saber qué interfaz desarrollar, realizamos una serie de *mockups*<sup>2</sup> para presentárselos a los profesores del centro, quienes podrían decidir cuál es el diseño y la utilidad más interesante o qué funcionalidad era la más apropiada para orientarnos a la hora de realizar la aplicación.

A continuación, se explican los *mockups* y las conclusiones obtenidas tras la reunión.

### 5.2.1. Mockup ejemplo 1: Click con el botón derecho.

Una vez activada la extensión, como mostramos en la figura 5.1, una de las ideas propuestas consistía en seleccionar la palabra o el texto que presente dificultades para el usuario pulsando click derecho posteriormente sobre ella. En el menú desplegado debemos seleccionar *mi-plugin*, correspondiente con la aplicación a desarrollar. Esto desplegaría un submenú que enumeraría los servicios prestados por el programa. El usuario podrá escoger la opción que necesite. En este ejemplo, clicando sobre “definiciones” podrá ver todas las acepciones de la palabra “perros” por medio de una ventana emergente.

---

<sup>2</sup>Fotomontajes que permiten a los diseñadores gráficos y web mostrar al cliente cómo quedarán sus diseños.

## Cuidados de tu perro o cachorro

Es importante que sepas cómo cuidar a tu perro. Para ayudarle, hemos reunido gran cantidad de información, desde el cuidado del pelo y las patas hasta consejos sobre su salud dental.

### El lenguaje corporal de tu perro

Una de las razones por las que los **perros** son tan buenos animales de compañía es por su capacidad para comunicarse con nosotros. De hecho, ellos suelen entendernos mejor a nosotros que nosotros leer su lenguaje corporal. Pero si sabes interpretar la postura, las orejas, la boca, el pelo y la cola de tu perro, podrás entenderlo mejor.

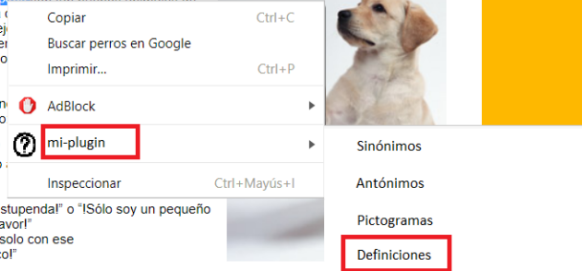
Sonrisa torcida: "¡Seamos amigos!"  
Labios retraídos para enseñar las encías: Igual que antes, pero con un gruñido.

Orejas levantadas: "¿Qué sucede?"  
Orejas retraídas o planas: "Necesito esto."

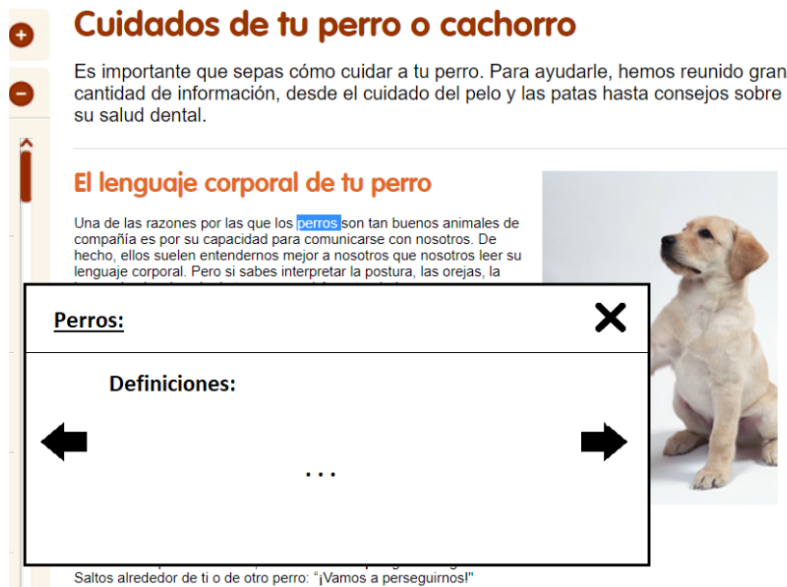
Ojos medio cerrados: "¡La vida es estupenda!" o "¡Sólo soy un pequeño cachorro, sé amable conmigo, por favor!"  
Ojos totalmente abiertos: "¡Déjame solo con ese cartero/gato/ardilla/gallina de plástico!"

Pata delantera levantada: "¡Vamos a jugar!"

Inclinación rápida de cabeza, con un ladrido: "¡Juega conmigo ahora!"  
Saltos alrededor de ti o de otro perro: "¡Vamos a perseguirnos!"



(a) Selección de los servicios que necesita el usuario



(b) Ventana emergente resultante

Figura 5.1: Selección de palabras pulsando botón derecho

### 5.2.2. Mockup ejemplo 2: Barra de menú.

Como segunda propuesta, expusimos la posibilidad de diseñar una barra de menú con una serie de servicios que ofrecería la aplicación. Ante esta situación, distinguimos dos posibilidades:

- **OPCIÓN 1: MENÚ SELECCIONANDO LA PALABRA.** El usuario debe seleccionar el texto de la página web en la que se encuentra que le presente dificultades y escoger una de las opciones existente en la barra de menú personalizada como se muestra en la figura 5.2, clicando la opción deseada. Posteriormente, aparecerá una ventana emergente con todos los significados de la palabra.



(a) Escogiendo la palabra seleccionada por el usuario



(b) Ventana emergente resultante

Figura 5.2: Selección de palabras a través de una barra de menú

- **OPCIÓN 2: ESCRIBIR EN EL MENÚ LA PALABRA.** Otra característica consistía en incluir una barra de búsqueda en el menú personalizado. De esta manera, el usuario tendría la posibilidad de escribir una palabra o texto que le presente problemas. Una vez introducida la palabra, en el ejemplo de la figura 5.3 deberíamos clicar en la opción “definiciones”. Nos aparecerá una ventana emergente con todas las acepciones de la palabra perros.

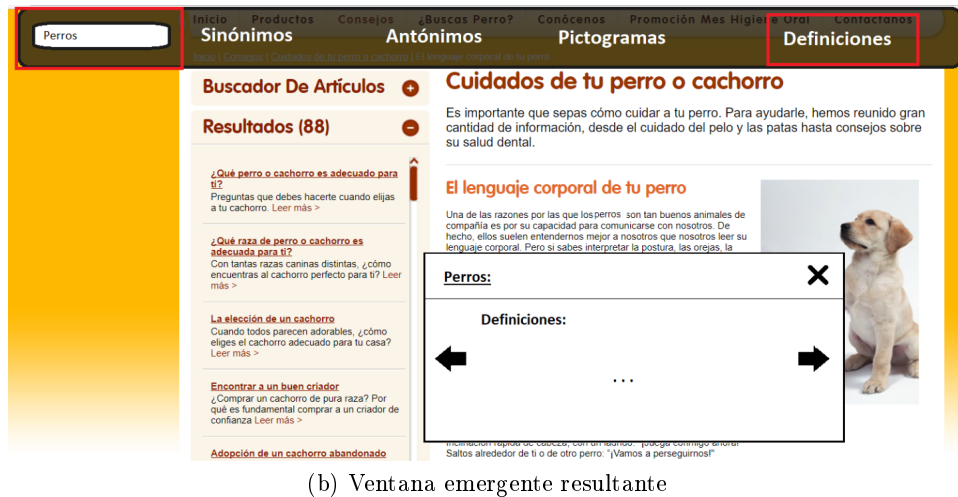
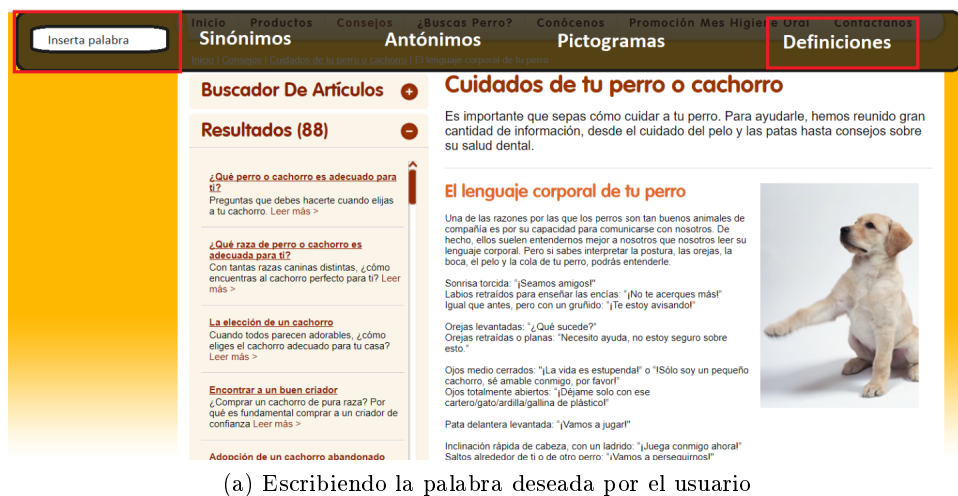


Figura 5.3: Selección de palabras a través de una barra de menú escribiendo texto

### 5.2.3. Menú guardar.

Una vez explicada la temática de la aplicación, también planteamos si podría resultar interesante poder almacenar los datos que los usuarios hayan buscado en algún momento, para que los puedan consultar siempre que quieran, como mostramos en la figura 5.4

### 5.2.4. Marcas de guardado en el documento.

Como última aportación a la explicación, concluimos mostrando una idea extraída de la aplicación *NavegaFácil*, que como mostramos en la figura 5.5 consiste en añadir marcas de guardado en todas aquellas palabras que el usuario haya buscado cualquier tipo de información.

### 5.2.5. Colegio Estudio 3 AFANIAS: Primeras conclusiones

Una vez terminada la exposición, fue momento de contrastar nuestras ideas con las de Carlos Roldán, el cual nos indicó las siguientes pautas que deberíamos tener en cuenta:

- Algunos de los alumnos sí pueden seleccionar texto, pero no todos.
- Recalcó la importancia de ampliar el texto.
- Algunos de los alumnos sí que podrían querer obtener información textual.
- Indicó que resultaría interesante poder marcar la información más importante a modo de resúmenes.
- En cuanto al menú, deben entender las palabras. Por ejemplo, en lugar de mostrar la opción “sinónimos” sustituirla por “palabras parecidas”. Otra opción sería mostrar pictogramas<sup>3</sup> en lugar de palabras.
- Sería buena idea poner los textos en mayúsculas.
- Añadir la opción a navegar por Youtube o la Wikipedia sería una gran aportación.
- Como cada alumno tiene distintos tipos de discapacidad, resulta interesante poder personalizar para cada usuario qué opciones tendrá disponibles en la extensión.

---

<sup>3</sup>Dibujo o signo gráfico que expresa un concepto relacionado materialmente con el objeto al que se refiere.



(a) Seleccionar el servicio que queremos



(b) Escoger la palabra que necesitamos



(c) Ventana emergente resultante

Figura 5.4: Obteniendo los valores guardados por el usuario

### El lenguaje corporal de tu perro

Una de las razones por las que los perros son tan buenos animales de compañía es por su capacidad para comunicarse con nosotros. De hecho, ellos suelen entendernos mejor a nosotros que nosotros leer su lenguaje corporal. Pero si sabes interpretar la postura, las orejas, la boca, el pelo y la cola de tu perro, podrás entenderle.

Sonrisa torcida: "¡Seamos amigos!"

Labios retraídos para enseñar las encías: "¡No te acerques más!"  
Igual que antes, pero con un gruñido: "¡Te estoy avisando!"

Orejas levantadas: "¿Qué sucede?"

Orejas retraídas o planas: "Necesito ayuda, no estoy seguro sobre esto."

Ojos medio cerrados: "¡La vida es estupenda!" o "¡Sólo soy un pequeño cachorro, sé amable conmigo, por favor!"

Ojos totalmente abiertos: "¡Déjame solo con ese cartero/gato/ardilla/gallina de plástico!"

Pata delantera levantada: "¡Vamos a jugar!"

Inclinación rápida de cabeza, con un ladrido: "¡Juega conmigo ahora!"

Salto alrededor de ti o de otro perro: "¡Vamos a perseguirnos!"

Aspecto erizado – orejas, rabo y pelo del cuello erizados todos ellos:

"¡Soy un perro de primera!"

Igual que antes, pero con un lento meneo del rabo y gruñidos: "¡No lo olvides!"



Figura 5.5: Ejemplo de marcas de guardado

En cuanto a los *mockups* presentados, seleccionó como mejor idea desarrollar un menú personalizado con la opción de poder escribir texto en una barra de búsqueda. Descartó la idea de generar marcas de guardado, ya que podría causar confusión.

Tras debatir la mejor manera de diseñar la aplicación, decidimos concertar una segunda reunión una vez realizado todo o casi todo el desarrollo de nuestra extensión para mostrárselo a los profesores de los alumnos para que vieran el funcionamiento de nuestra extensión y nos indicaran si cubría sus necesidades. Posteriormente, en una tercera visita al centro, los alumnos podrían utilizar la extensión y conocer sus opiniones.

### 5.3. Colegio Estudio 3 AFANIAS: Segunda reunión

Tras la fase de desarrollo de la aplicación (ver capítulos 6 y 7), realizamos una segunda reunión con el centro la primera semana de mayo con Juan Miguel Fernández (tutor de los alumnos con los que íbamos a realizar la evaluación con usuarios finales).

Uno de los objetivos de la reunión consistía en exponer nuestra herramienta para conocer si la aplicación cubría las necesidades de sus alumnos. La respuesta de Juan Miguel Fernández resultó muy positiva, indicándonos que nuestro plugin reunía muchas funcionalidades que serían de gran ayuda para sus alumnos. Además, nos indicó también que incluir en las opciones de nuestra herramienta el blog del centro resultaría útil durante sus clases.

Otro objetivo fue marcar la fecha para realizar la prueba con los alumnos del colegio, fijada para la segunda semana de mayo.

## 5.4. Colegio Estudio 3 AFANIAS: Tercera reunión

El objetivo de esta reunión consistió en probar nuestra aplicación con los usuarios finales, en este caso diez alumnos del colegio Estudio 3 AFANIAS. Esta visita se dividió en varias partes:

- Mostrar la aplicación a José Ramón Díaz, profesor responsable de los alumnos en el aula de informática.
- Realizar una presentación en el aula de los alumnos mostrándoles por primera vez la aplicación y enseñándoles cómo funciona. Para ello, preparamos una serie de actividades en las que cubríamos toda la funcionalidad de nuestra aplicación. Por ejemplo, para mostrar la funcionalidad de Youtube, como Juan Miguel Fernández nos indicó que en su viaje de fin de curso irían a Port Aventura, les mostramos vídeos en Youtube sobre las atracciones del parque.
- Probar nuestra herramienta en el aula de informática durante una hora siguiendo las pautas de su profesor. Allí, les indicó distintas actividades para probar la aplicación: buscar en internet sus películas favoritas para leer su sinopsis y posteriormente buscar la política de privacidad de *WhatsApp*, con el fin de comprender sus contenidos.

Finalmente, los alumnos y profesores rellenarían unos cuestionarios que preparamos para conocer cuáles eran sus opiniones sobre la aplicación.

### 5.4.1. Colegio Estudio 3 AFANIAS: Cuestionarios sobre la aplicación

Como trabajo previo a la reunión, creamos una serie de cuestionarios orientados a los distintos perfiles de los usuarios con los que trabajaríamos: profesores y alumnos. Con estos cuestionarios queríamos conseguir las opiniones personales de todas las personas que usaron nuestra herramienta. De esta manera, conoceríamos como de útil y fácil les resultó nuestro plugin.



Para ello, creamos una estructura tipo test de preguntas y respuestas de *tipo semáforo con emoticonos*, como vemos en la figura 5.6, donde el verde representaba *bien*, el amarillo *regular* y el rojo *mal*.

Además, en el caso de los alumnos dejamos una última pregunta donde pudieran expresar su opinión de manera escrita. En el caso del cuestionario de los profesores, formulamos una serie de cuestiones cuya respuesta pudieran desarrollar de manera escrita.



Figura 5.6: Respuestas en forma de semáforo

#### 5.4.1.1. Cuestionarios sobre la aplicación para los usuarios

Las preguntas formuladas para los alumnos son:

- ¿Te ha ayudado la opción de Definiciones?
- ¿Te ha ayudado la opción de Palabras parecidas?
- ¿Te ha ayudado la opción de Palabras distintas?
- ¿Te ha ayudado la opción de Pictogramas?
- ¿Te ha ayudado la opción de YouTube?
- ¿Te ha ayudado la opción de Wikipedia?
- ¿Te ha ayudado la opción de Resumen?
- ¿Te ha ayudado la opción de Blog del cole?
- ¿Te ha gustado?
- ¿Te ha ayudado?
- ¿Lo usarías para navegar por internet?
- ¿Algo más que nos quieras decir?

#### 5.4.1.2. Cuestionarios sobre la aplicación para los profesionales

Las preguntas formuladas para los profesores son:

- ¿Te ha ayudado la opción de Definiciones?
- ¿Te ha ayudado la opción de Palabras parecidas?
- ¿Te ha ayudado la opción de Palabras distintas?
- ¿Te ha ayudado la opción de Pictogramas?
- ¿Te ha ayudado la opción de YouTube?
- ¿Te ha ayudado la opción de Wikipedia?
- ¿Te ha ayudado la opción de Resumen?
- ¿Te ha ayudado la opción de Blog del cole?
- ¿Crees que les ha gustado?
- En general, ¿crees que les ha ayudado?
- ¿Lo usarías en el aula?
- Si sí lo usarías en el Aula, ¿nos dices cómo?
- ¿Echas en falta alguna opción?
- Comentarios adicionales.

#### 5.4.2. Colegio Estudio 3 AFANIAS: Resultados tras la tercera reunión

Una vez concluidas todas las actividades preparadas con los usuarios tras la reunión, recogimos los cuestionarios realizados por los usuarios. Como podemos ver en la tabla 5.1, donde recogemos las respuestas de los cuestionarios que respondieron los usuarios, nuestra aplicación obtuvo muy buena acogida por parte de los alumnos. Estudiando las respuestas, podemos afirmar que:

- La funcionalidad proporcionada por nuestra aplicación es la adecuada, ofreciéndoles parte de la ayuda que necesitan para enfrentarse a las dificultades que les puede ocasionar la lectura en la red.
- La aplicación les ha ayudado.
- Nuestra herramienta les ha gustado.
- Todos los alumnos contestaron que la usarían para navegar por internet.

- Los comentarios ofrecidos por los alumnos muestran mucha gratitud y afirman que les ayudará para navegar por internet y aprender.

Unos días después de la actividad, recibimos las respuestas a los cuestionarios realizadas por los profesores. En la tabla 5.2 recogemos sus respuestas, tras las que podemos afirmar que:

- Todas las funcionalidades presentadas por la aplicación les parecen de ayuda. En el caso de los pictogramas, un profesor expone que le resulta de menos ayuda, ya que opina que esta funcionalidad tiene más sentido en el apoyo adulto, no para alumnos en clase.
- Varios profesores indicaron que sería mejor idea llamar a los antónimos como *Palabras contrarias* en lugar de *Paralabras diferentes* como lo habíamos nombrado en nuestra aplicación.
- Todos los profesores opinan que les ha gustado a los alumnos.
- Los profesionales aseguran que la herramienta ha ayudado a los alumnos.
- Todos los profesores confirman que usarán nuestra aplicación a la hora de impartir sus clases.
- Los profesores expusieron que lo usarían como apoyo a la lectura y visualización de temas curriculares en Youtube. Indicaron que la función de resumen les sería de gran utilidad. Además, apuntaron que con la aplicación se conseguía el objetivo que los profesionales se planteaban: que los alumnos sean críticos con la información que se les plantea en internet.
- Indican que es una herramienta muy buena. Valoran de manera muy positiva la buena acogida de los alumnos.
- Un profesor indica que tiene ganas de probarlo en clases con sus alumnos, al igual que el resto de docentes que han visto la aplicación pero no han tenido ocasión de probarla con sus alumnos.

Además de la información recogida en los cuestionarios, en las aulas durante la presentación de la aplicación y cuando los alumnos probaron nuestra herramienta, pudimos comprobar la gran acogida que presentó por su parte. Los alumnos se mostraron ilusionados y con ganas de aprender a usarla. Mientras la probaban por ellos mismos, comprobamos que les resultaba fácil de usar y que les proporcionaba ayuda ante los textos y palabras que no entendían.

Concluimos la valoración de la aplicación con resultados muy positivos, tanto de cara a la usabilidad como la utilidad de la aplicación.

Alumnos	Definiciones	Palabras parecidas	Palabras contrarias	Pictogramas	Youtube	Wikipedia	Resumen	Blog del cole	¿Te ha gustado?	¿Te ha ayudado?	¿Lo usarías para navegar por internet?	Comentarios
Alumno 1	Bien	Bien	Bien	Bien	Bien	Bien	Bien	Bien	Bien	Bien	Bien	Me gustaría ver opciones de páginas relacionadas
Alumno 2	Bien	Bien	Bien	Bien	Bien	Bien	Bien	Bien	Bien	Bien	Bien	-
Alumno 3	Bien	Bien	Bien	Bien	Bien	Bien	Bien	Bien	Bien	Bien	Bien	-
Alumno 4	Bien	Bien	Bien	Bien	Bien	Bien	Bien	Bien	Bien	Bien	Bien	Que muchas gracias, gracias por todo, me ha gustado mucho la actividad
Alumno 5	No utilizado	No utilizado	No utilizado	Bien	Bien	No utilizado	No utilizado	Bien	Bien	Bien	Bien	-
Alumno 6	Bien	Bien	Bien	Regular	Bien	Bien	Bien	Bien	Bien	Bien	Bien	Me ha gustado buscar películas que nos gusten
Alumno 7	Bien	Bien	Bien	Bien	Bien	Bien	Bien	Bien	Bien	Bien	Bien	Muchísimas gracias por todo. Así me has ayudado mucho y así aprendo más
Alumno 8	No utilizado	No utilizado	No utilizado	No utilizado	Bien	Bien	Bien	No utilizado	Bien	Bien	Bien	-
Alumno 9	Bien	Mal	Mal	Mal	Mal	Mal	Mal	Regular	Bien	Regular	Bien	La aplicación puede ser útil.
Alumno 10	Bien	Bien	Bien	Bien	Bien	Bien	Bien	Bien	Bien	Bien	Bien	La aplicación me ha molado un montón. Me ha parecido muy fácil.

Tabla 5.1: Respuestas del cuestionarios de los alumnos

Profesor	Definiciones	Palabras parecidas	Palabras contrarias	Programas	Youtube	Wikipedia	Resumen	Blog del cole	¿Crees que les ha gustado?	¿Crees que les ha ayudado?	¿Lo usarías en el aula?	¿Como y para qué lo usarías en el aula?	¿Edas alguna opción en falta?	Comentarios
Profesor 1	Bien	Bien	Bien (Lo cambiaría por Palabras contrarias)	Bien	Bien	Bien	Bien	Bien	Bien	Bien	Sí	Apoyo a la lectura comprensiva. Visualización en Youtube sobre temas curriculares y centros de interés. Resumen de textos largos con los que van a trabajar los alumnos	Me imagino que durante el uso con los alumnos irán saliendo ideas	Muy buena herramienta, sobre todo para el trabajo en el colegio. Valoro positivamente su transversalidad y la gran adaptación por parte de los alumnos
	Bien	Bien	Bien (Lo cambiaría por Palabras contrarias)	Bien	Bien	Bien	Bien. Minuto. La búsqueda de información es lo que más les cuesta por la comprensión y el análisis de textos habitualmente largos y esta opción les ayuda bastante.	Bien	Bien. Su actitud y positivo vismo lo confirman	Bien	Sí	El objetivo planteado de conseguir que sean críticos con la información a la que acceden se ve más fácil de conseguir con esta herramienta.	Adaptación a alumnos visuales (pictogramas, microfilm), a cifras romanos o árabes.	Muy positiva y con ganas de probarla. Los complementos que se le ha mostrado opinan igual también.
Profesor 2	Bien	Bien	Bien (Lo cambiaría por Palabras contrarias)	Bien	Bien	Bien	Bien	Bien	Bien	Bien	Sí	Regular. Al ser una herramienta principalmente para lectoescritores, esta opción es más para apoyo del adulto que para el alumno en clase	Muy positiva y con ganas de probarla. Los complementos que se le ha mostrado opinan igual también.	
Profesor 3	Bien	Bien	Bien	Bien	Bien	Bien	Bien	Bien	Bien	Bien	Sí	Sobre todo la opción de sinónimos, resumen y el acceso a Youtube.	No	.

Tabla 5.2: Respuestas del cuestionarios de los profesores



## Capítulo 6

# Descripción de la aplicación

En este capítulo realizaremos un recorrido de la aplicación explicando las distintas funcionalidades que podemos encontrar en nuestra extensión: ReadIt.

### 6.1. Barra de menú de la aplicación

La barra horizontal de la figura 6.1 es el elemento principal de nuestra aplicación. Una vez activado nuestro plugin, este menú permanecerá visible en la parte superior de todas las páginas por las que el usuario navegue.

En ella disponemos de todas las funcionalidades que nos ofrece la aplicación, las cuales iremos explicando de izquierda a derecha acorde con la figura 6.1

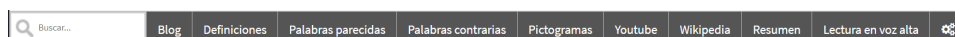


Figura 6.1: Barra principal de la aplicación

#### 6.1.1. Barra de búsqueda

Como ya sabemos, el usuario dispone de dos maneras de buscar el texto que quiere. Por un lado, puede seleccionar la palabra o frase que se encuentra en la página web. De otra manera, puede hacer uso de la barra principal que ofrece nuestra aplicación, insertando el texto en la barra de búsqueda.

### 6.1.2. Blog

A través de esta opción, podemos acceder al Blog del colegio Estudio 3 AFANIAS<sup>1</sup>, el cual se nos presentará en una ventana nueva. Incluimos el blog para el estudio en el centro con los usuarios finales, ya que aportaría ayuda a los docentes, sin embargo, esta opción es configurable.

### 6.1.3. Definiciones

Esta opción nos ofrece las distintas acepciones de la palabra escogida por el usuario. Podemos encontrar casuísticas distintas.

Por un lado, supongamos que el usuario ha buscado la palabra *casa*. El resultado sería el que muestra la figura 6.2

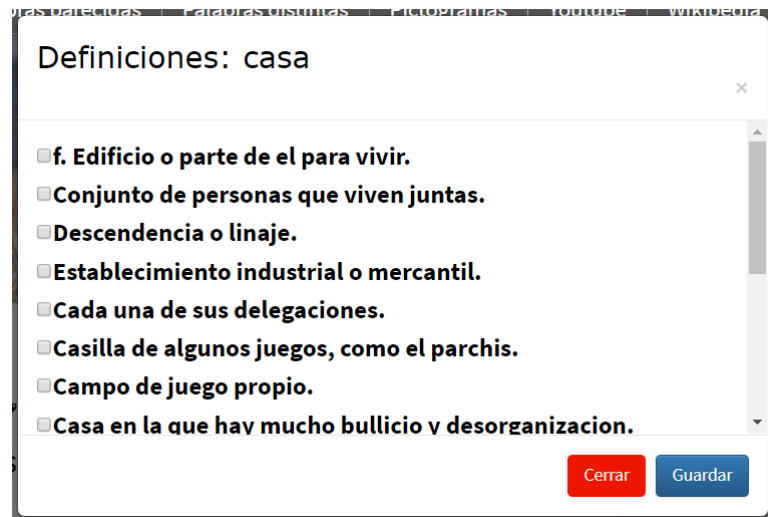


Figura 6.2: Definiciones de la palabra casa

De otro modo, cabe la posibilidad de que el usuario haya escrito mal la palabra o dicha palabra no exista, por lo que el resultado será el que presenta la figura 6.3

### 6.1.4. Palabras parecidas

Esta opción nos ofrece la posibilidad de mostrar al usuario los sinónimos de la palabra que ha seleccionado. Usamos la nomenclatura *Palabras parecidas* para facilitar a los usuarios su significado. De nuevo, encontramos dos casuísticas, que el usuario haya buscado una palabra correcta o incorrecta. El resultado es el mismo que para las definiciones.

<sup>1</sup><http://colegioestudio3.blogspot.com.es/>



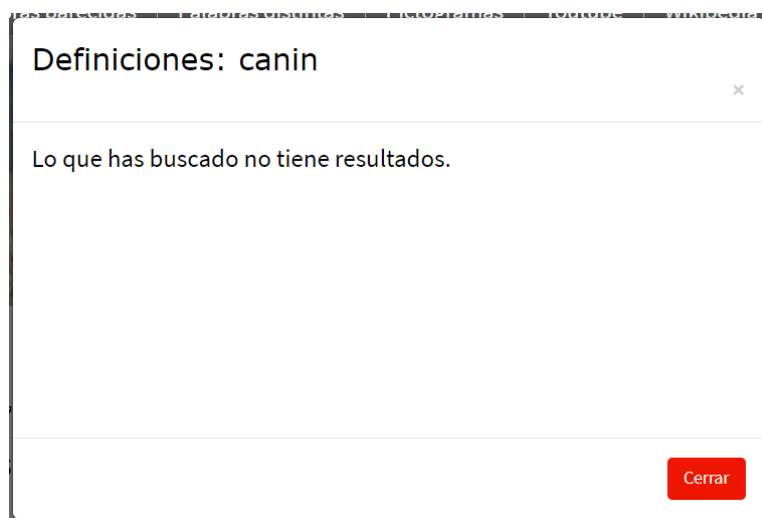


Figura 6.3: Resultado de una palabra que no presenta definiciones

#### 6.1.5. Palabras contrarias

Esta opción nos ofrece la posibilidad de mostrar al usuario los antónimos de la palabra que ha seleccionado. Usamos la nomenclatura *Palabras contrarias* para facilitar a los usuarios su significado. De nuevo, encontramos dos casuísticas, que el usuario haya buscado una palabra correcta o incorrecta. El resultado es el mismo que para las definiciones.

#### 6.1.6. Pictogramas

Por medio de esta opción, ofrecemos a los usuarios la posibilidad de ver el texto seleccionado en pictogramas. De esta manera, como vemos en la figura 6.4, mostramos los pictogramas asociados a la palabra *casa*. En caso de no existir un pictograma para la palabra o frase escogida, mostramos un mensaje de error.

#### 6.1.7. Youtube

A través de esta opción, podemos acceder a *Youtube*, abriéndose en una nueva pestaña. Cabe la posibilidad de abrir una nueva ventana de *Youtube* simple, es decir, sin ninguna búsqueda realizada. En el caso de que un usuario haya seleccionado algún texto, la ventana nueva se abrirá con la búsqueda de la palabra seleccionada.



Figura 6.4: Pictogramas de la palabra casa

#### 6.1.8. Wikipedia

A través de esta opción, podemos acceder a *Wikipedia*, abriéndose en una nueva pestaña. Cabe la posibilidad de abrir una nueva ventana de *Wikipedia* sin ninguna búsqueda realizada. Si el usuario ha seleccionado algún texto, la ventana nueva abrirá la *Wikipedia* con la búsqueda de la palabra seleccionada.

#### 6.1.9. Resumen

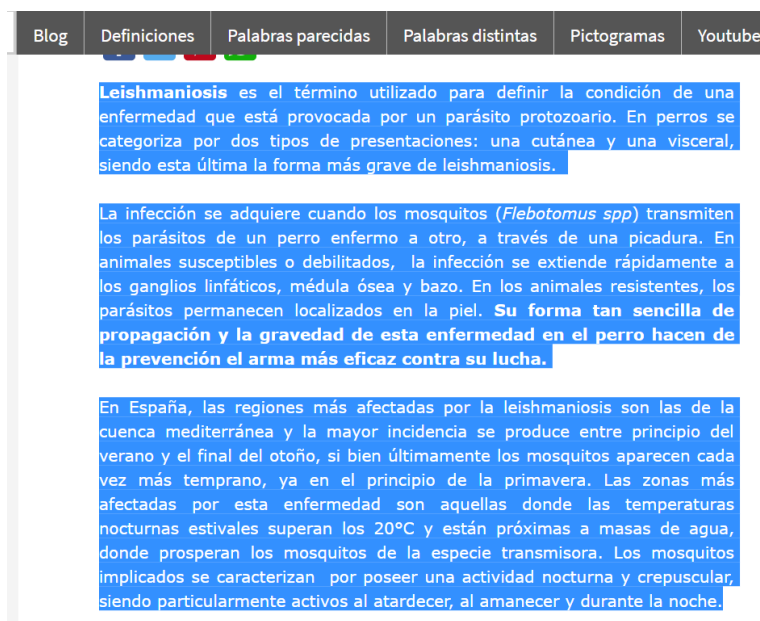
Gracias a esta opción el usuario podrá leer el resumen de un texto largo que le dificulte su comprensión. Como mostramos en la figura 6.5, el usuario ha de seleccionar el texto que le cause problemas de comprensión y escoger la opción de *Resumen*.

#### 6.1.10. Lectura en voz alta

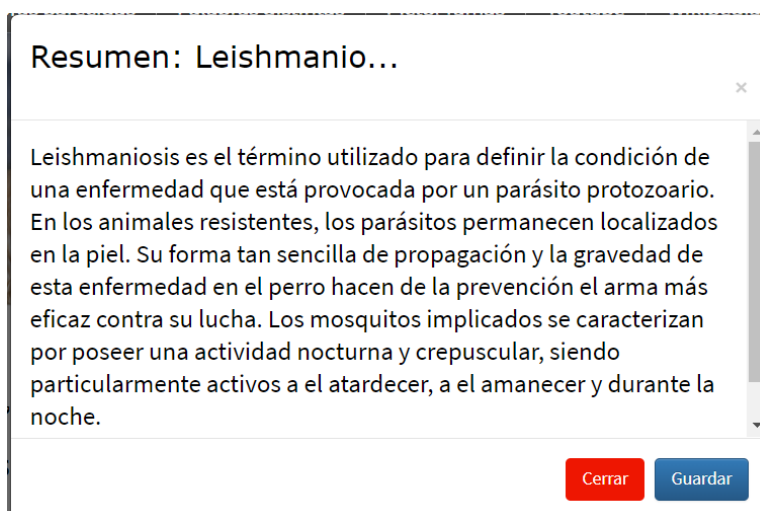
Usando esta opción, el usuario podrá escuchar en voz alta el texto que ha seleccionado o escrito en la barra de búsqueda.

#### 6.1.11. Icono de los engranajes

Como mostramos en la figura 6.6, seleccionando esta opción el usuario podrá ver todas y cada una de las búsquedas que ha realizado, así como los elementos que haya seleccionado en dicha búsqueda. Más adelante en la sección 6.3 explicaremos cómo podemos guardar los datos.



(a) Texto seleccionado que queremos resumir



(b) Resumen del texto seleccionado

Figura 6.5: Ejemplo usando la opción de resumen

## 6.2. Configuración de la barra de menú de la aplicación

La barra horizontal presenta todos los servicios que nos ofrece la aplicación. Sin embargo, es posible que no todos sean de utilidad dependiendo del usuario que lo use. Por ello, implementamos una barra configurable en

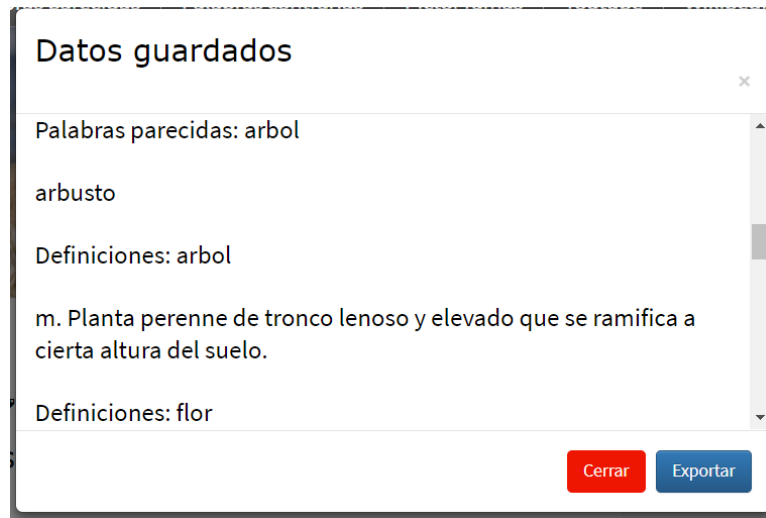


Figura 6.6: Ejemplo de los datos guardados por el usuario

la que un tutor del usuario podrá adaptarla acorde con sus necesidades.

Supongamos que solo se precisa de los servicios *Definiciones*, *Palabras parecidas*, *Youtube* y *Lectura en voz alta*. El usuario podrá clicar en nuestro plugin, abriéndose un menú donde podrá seleccionar las opciones que desee, siendo el resultado que vemos en la figura 6.7.

### 6.3. Guardar los datos del usuario

Otra funcionalidad que proporciona nuestra aplicación es la posibilidad de guardar los datos del usuario según las búsquedas que haya realizado. Para ello, el usuario ha de buscar las palabras que le presenten problemas. Por ejemplo, como vemos en la figura 6.8, suponemos que busca los sinónimos de *árbol*, selecciona la primera palabra y pulsa el botón *Guardar*. Además desea conocer cuáles son las definiciones de la palabra *árbol* y selecciona la primera acepción y pulsa el botón *Guardar*.

Una vez realizadas las búsquedas pertinentes, el usuario podrá ver sus datos guardados pulsando en la opción de los engranajes, como explicamos la sección 6.1.11. El resultado será el de la figura 6.9, en la que podemos ver en color rojo los datos relativos a la búsqueda de las definiciones de la palabra *casa* con las dos acepciones que escogió el usuario y en color naranja la búsqueda del pictograma *mosquito*.

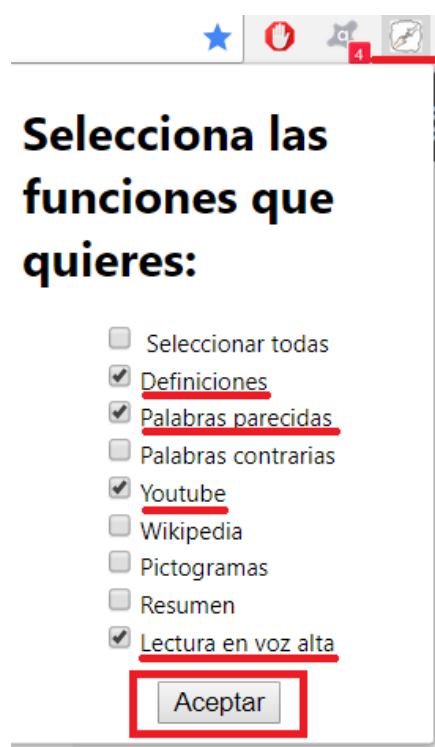
Además, los datos guardados por el usuario se almacenan de manera persistente en el navegador. Es decir, si el usuario termina la sesión apagando el ordenador o saliendo del navegador, la próxima vez que navegue por las

páginas sus datos permanecerán guardados.

## 6.4. Exportar los datos del usuario

Nuestra aplicación dispone de la posibilidad de exportar los datos que el usuario ha guardado. Resulta interesante poder tener en un fichero todas las búsquedas que el usuario haya realizado, con la finalidad de repasar en casa los conceptos estudiados en clase o para saber cuáles son los términos que presentan mayor dificultad para el usuario.

Para poder exportar los datos, debemos tener abierta la ventana con los datos guardados del usuario. Como vemos en la figura 6.10 debemos pulsar el botón *exportar*, lo que nos generará un archivo con el nombre *misDatosGuardados.html* donde se exportarán todos los elementos guardados del usuario



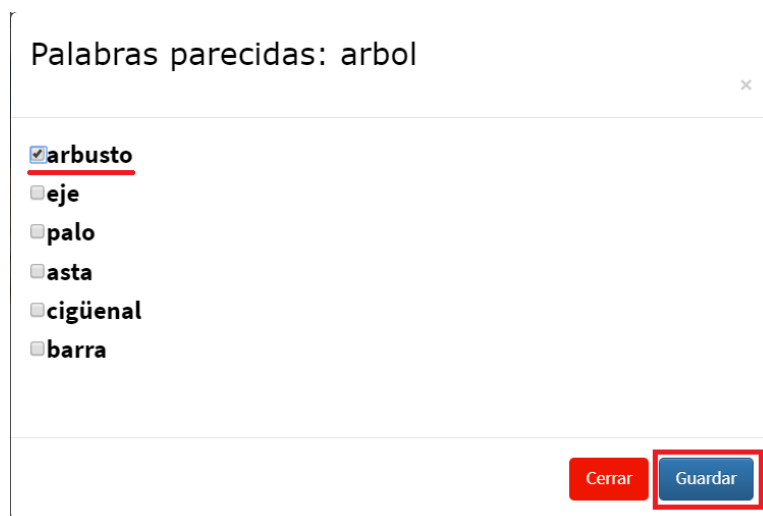
(a) Escogiendo las opciones a mostrar en la barra de menú



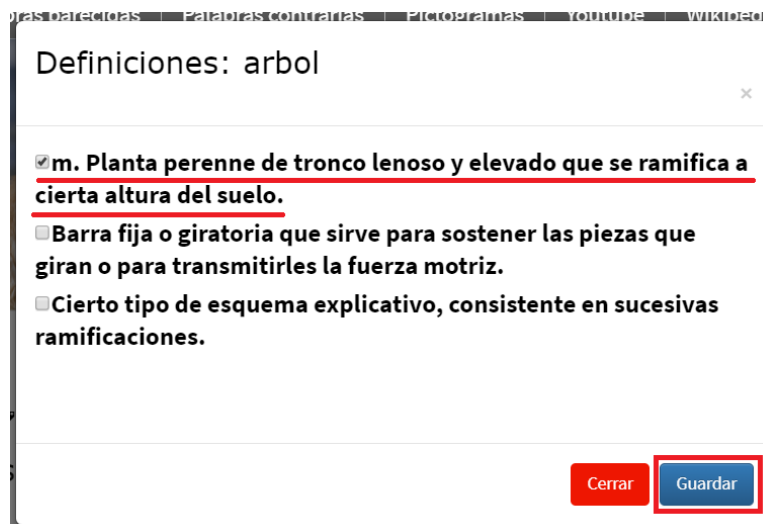
(b) Resultado de filtrar las opciones deseadas

Figura 6.7: Resultado de filtrar los servicios adecuados para el usuario

con una apariencia sencilla.



(a) Seleccionando los sinónimos de la palabra árbol



(b) Buscando las acepciones de la palabra árbol

Figura 6.8: Guardando los datos del usuario

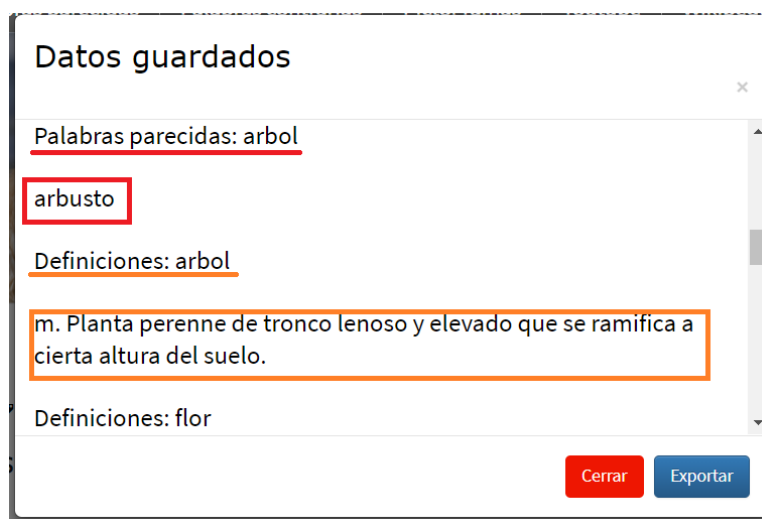
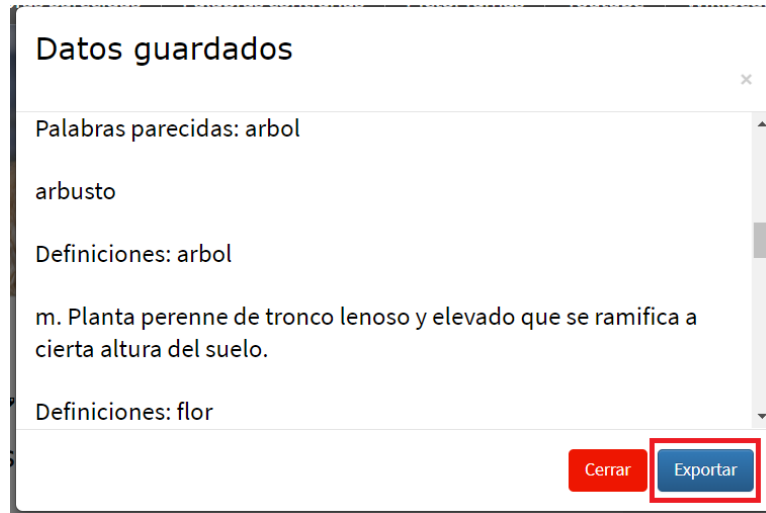


Figura 6.9: Ejemplo de los datos guardados por el usuario



(a) Seleccionando la opción de exportar datos

---

Palabras parecidas: arbol

arbusto

Definiciones: arbol

m. Planta perenne de tronco lenoso y elevado que se ramifica a cierta altura del suelo.

Definiciones: flor

f. En las plantas espermafitas, estructura donde estan los organos de reproduccion.

Definiciones: juego

Palabras parecidas: casa

hogar

Definiciones: casa

f. Edificio o parte de el para vivir.

Palabras parecidas: gato

minino

(b) Fichero html exportado con los datos guardados

Figura 6.10: Guardando los datos del usuario



## Capítulo 7

# Arquitectura de la aplicación

Este capítulo lo dedicaremos a estudiar la estructura y arquitectura de nuestra extensión para Chrome, a la que hemos dado el nombre de ReadIt. En primer lugar explicaremos a alto nivel cómo están ordenados los ficheros que lo componen. De esta manera, en un segundo vistazo desarrollaremos en profundidad su contenido y funcionalidad.

Además, explicaremos detalladamente como podemos hacer que nuestro proyecto sea extensible acorde a las nuevas necesidades que los usuarios puedan necesitar.

### 7.1. Introducción

En el capítulo 4 desarrollamos en detalle cómo está formada la estructura de una extensión de Google Chrome. Dedicaremos esta sección en introducir brevemente la organización de los ficheros que componen el diseño de nuestro plugin.

Como vemos en la figura 7.1, podemos distinguir en la capa más alta:

- El fichero de configuración *manifest.json*.
- El archivo *popup.html*, que representa la *acción del navegador* de nuestra extensión.
- El fichero *style.css*, que se corresponde con uno de los *scripts de contenido*, en el que guardamos los estilos que deseamos dar a la parte visual de nuestra extensión.
- El archivo *background.js*, que representa el *script de fondo* de la aplicación.

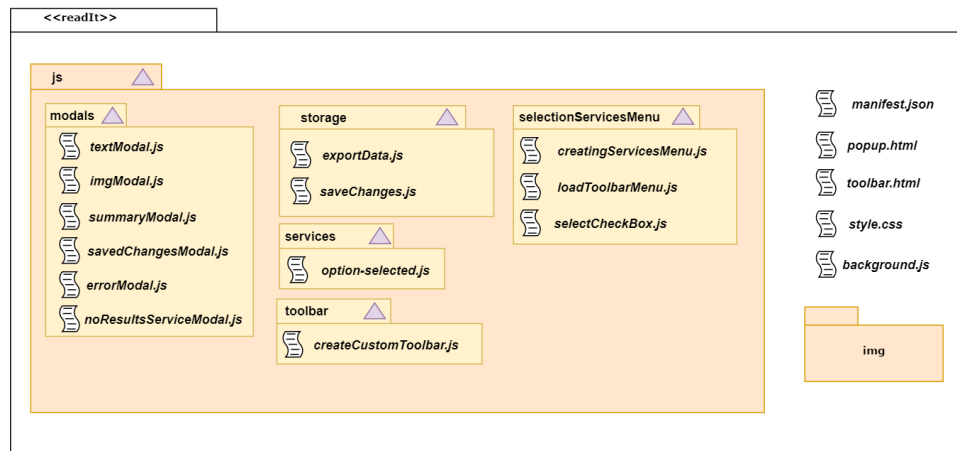


Figura 7.1: Estructura del proyecto

- El fichero *toolbar.html*, que representa la barra de menú principal de la aplicación.
- La carpeta *js*, donde reunimos todos los ficheros almacenados en distintas carpetas para proporcionar la funcionalidad del proyecto. Corresponderán a los *scripts de contenido*.
- La carpeta *img*, en la que se encuentran todas las imágenes que presenta el plugin, como por ejemplo su imagen principal.

## 7.2. Arquitectura

En esta sección explicaremos en detalle la función de cada uno de los ficheros que componen la aplicación siguiendo una vez más la imagen de la figura 7.1.

Estudiaremos los ficheros esenciales para la aplicación y todo el contenido que compone el directorio *js*.

### 7.2.1. Archivos principales

Los componentes principales de la extensión son:

- *STYLE.CSS*: En este fichero generaremos los estilos que queremos proporcionar a nuestro plugin. En concreto, daremos forma a la barra de herramientas que mostramos en la página y al selector de servicios ofrecidos de nuestra extensión. Se trata de un fichero obligatorio.

- MANIFEST.JSON: Este es el archivo de configuración donde almacenaremos todos los datos necesarios para que la extensión funcione en nuestro navegador.
  - Guardaremos la información relativa al nombre y descripción de nuestro proyecto así como el autor.
  - Añadiremos las imágenes que identifican a nuestra aplicación.
  - Incluiremos todos los ficheros que necesitamos para realizar la funcionalidad que queremos.
  - Indicaremos en qué tipo de páginas funcionará nuestra extensión. En nuestro caso, se aplicará en todas aquellas que contengan la signatura *http* y *https*.
  - Añadiremos los permisos de los que dispondrá nuestra extensión. Esto es importante por dos motivos: para poder llamar a los servicios que la API de Chrome nos ofrece, como por ejemplo, *storage*, que nos permite guardar información en nuestro plugin. Además nos permite proporcionar a nuestra aplicación solo los permisos que nuestro proyecto requiere.
- POPUP.HTML: Con este documento definimos la vista en la que mostraremos los distintos servicios que ofrece nuestra aplicación llamando a los archivos que pertenecen a la carpeta *selectionServicesMenu* de la que hablaremos más adelante.
- TOOLBAR.HTML: En este fichero creamos nuestra barra de herramientas personalizada, con la que el usuario podrá hacer uso de los servicios que ofrece la aplicación. En un primer punto, se mostrarán unos servicios cargados por defecto hasta que el usuario la personalice acorde a las necesidades de la persona que la usará.

### 7.2.2. SelectionServicesMenu

En esta sección explicaremos el uso de los ficheros que contienen el directorio de *selectionServicesMenu*. Su importancia incide en configurar la vista *popup.html* nombrada anteriormente así como darle funcionalidad.

- CREATINGSERVICESMENU.JS: En este documento encontramos dos partes importantes:
  - Una lista que contiene los servicios que proporciona la aplicación. Cada uno de ellos formado por su identificador y nombre que veremos en nuestra barra de herramientas.
  - Una función *creatingServicesMenu.js* en la que generamos de manera dinámica la vista *popup.html* ya mencionada antes, en la que

por cada elemento de nuestra lista creamos un nuevo *checkbox*. Esto permite al usuario escoger los servicios que mejor se adapten a sus necesidades. Finalmente, añadimos el botón *Aceptar* para aplicar los cambios en nuestra barra de herramientas.

- `LOADTOOLBARMENU.JS`: La finalidad de este fichero es una de las más importantes que nos encontraremos. Es el encargado de comunicar nuestra aplicación con los elementos del árbol DOM de la página web en la que estamos navegando. Su función consiste en guardar los servicios que ha seleccionado el usuario en un array y enviarlos a la estructura encargada de gestionar la barra de herramientas personalizada.
- `SELECTCHECKBOX.JS`: En este archivo llamamos en primer lugar a la función que se encuentra en el fichero *creatingServicesMenu.js* para así tener los elementos de nuestra vista creados. A continuación, con ayuda de jQuery creamos la funcionalidad para conocer los elementos que han sido seleccionados. Además, incluye la opción de seleccionarlos todos en caso de necesitarlos.

### 7.2.3. Services

En esta sección hablaremos del contenido de la carpeta *services*. En ella encontraremos el fichero llamado *option-selected.js*, el cual es el encargado de conocer cuál es el texto que ha solicitado el usuario, ya sea seleccionado o escrito en nuestra barra de herramientas. Otra de sus funciones es proporcionar el servicio que el usuario haya solicitado escogiendo un elemento de la barra de herramientas.

Como el usuario puede escoger el texto deseado de dos maneras, existen dos formas de obtenerlo. En primer lugar, para conocer el texto que el usuario ha seleccionado, hacemos uso de la llamada *addEventListener* que nos proporciona JavaScript para saber si el usuario ha seleccionado una palabra o no. En esta llamada, nos apoyamos en la función *getSelectionText*, que hemos desarrollado para conocer si se ha seleccionado texto o no. En caso afirmativo, lo guardamos en una variable global.

A continuación, con ayuda de jQuery y su función *click*, podemos conocer cuándo el usuario pulsa cualquiera de los servicios que ofrece nuestra barra de herramientas. Dentro de la función, declaramos una llamada a la función *getSearchText*, en la que comprobamos si el usuario ha escrito algo en el buscador de la barra de herramientas. En caso afirmativo, se guarda en la variable global anteriormente mencionada y se queda el valor del texto seleccionado (si lo hubiera).

Posteriormente, creamos una variable *url* en la que almacenaremos la llamada que nos muestra el servicio deseado.

Finalmente, llamaremos a la función *callingWebService* en la que pasaremos como parámetros la url y el texto deseado por el usuario. Existen dos casos:

- En caso de no haber guardado nada en la variable global, es decir, el usuario no ha buscado ni seleccionado ningún texto o palabra, y pulse algún servicio ofrecido, mostraremos un mensaje informativo indicando al usuario que debe insertar o seleccionar el texto que desea buscar.
- En el caso de que el usuario sí ha proporcionado un texto, realizamos una llamada *GET* o *POST* dependiendo del servicio seleccionado, utilizando jQuery y guardamos el resultado parseado en una variable. En último lugar, llamaremos a una función que se encuentra en otro fichero (que explicaremos posteriormente) que nos mostrará el resultado en una ventana nueva.

Dependiendo del servicio seleccionado por el usuario, las funciones de parseado se muestran más o menos complejas. En el caso de las *definiciones*, *sinónimos* y *antónimos*, la manera de parsear se basa únicamente en recorrer el *JSON* recibido por el servicio.

El caso de parsear el servicio de los *pictogramas* requiere un proceso más complejo. Este servicio devuelve un array de *strings* con el siguiente formato: "*0 noun[2317, 6964] casa*", donde los números 2317 y 6946 representan los *ids* que guardan la imagen que se corresponde con la palabra buscada por el usuario, en este ejemplo, "casa". En el caso de buscar una frase tendría el siguiente formato: "*0 noun[6009, 6522] hola*", "*0 noun[2497, 24234] papá*", donde en cada *string* se guarda la correspondencia de cada palabra con sus *ids* asociados, en este ejemplo tras la búsqueda de "hola papá". Una vez obtenemos la información del array, debemos recorrerlo para guardar los *ids*, que son la información que precisamos para mostrar las imágenes. Cada vez que obtenemos una posición del array, recogemos un *string* que debemos parsear. Cuando hemos guardado los *ids* parseados de las imágenes que se corresponden con la palabra seleccionada en un array, los mostramos con el fichero *imgModal.js*, quien es el encargado de mostrar las imágenes del servicio de los *pictogramas*.

En cuanto al servicio de *resúmenes*, no precisa de ningún tipo de parseado, ya que la llamada *POST* devuelve un *string* con el resumen del texto seleccionado.

### 7.2.3.1. Servicios web

La aplicación ofrece los siguientes servicios, todos basados en servicios web externos:

- **DEFINICIONES.** Con esta opción mostramos todas las acepciones de la palabra seleccionada por el usuario. Para conseguirlas, la aplicación hace uso del siguiente servicio:

```
1 https://sesat.fdi.ucm.es/tfgapi/servicios/rest/  
   definicion/json/palabra_seleccionada
```

- **SINÓNIMOS.** Esta opción muestra los sinónimos de la palabra seleccionada por el usuario. Se corresponde con la opción *Palabras parecidas* de la barra de menú de la aplicación. Para conseguirlos, la aplicación hace uso del siguiente servicio:

```
1 https://sesat.fdi.ucm.es/tfgapi/servicios/rest/sinonimos  
   /json/palabra_seleccionada
```

- **ANTÓNIMOS.** Esta opción muestra los antónimos de la palabra seleccionada por el usuario. Se corresponde con la opción *Palabras contrarias* de la barra de menú de la aplicación. Para conseguirlos, la aplicación hace uso del siguiente servicio:

```
1 https://sesat.fdi.ucm.es/tfgapi/servicios/rest/antonimos  
   /json/palabra_seleccionada
```

- **PICTOGRAMAS.** Esta opción muestra los pictogramas de la palabra seleccionada por el usuario. Para conseguirlos, la aplicación hace uso del siguiente servicio:

```
1 https://sesat.fdi.ucm.es/serviciopictar/  
   palabra_seleccionada  
2 http://hypatia.fdi.ucm.es/conversor/Pictos/  
   id_palabra_seleccionada
```

- **RESÚMENES.** Esta opción muestra el resumen del texto seleccionado por el usuario. Para conseguirlo, la aplicación hace uso del siguiente servicio:

```
1 https://sesat.fdi.ucm.es/grafeno/run/summary_es
```

- **LECTURA EN VOZ ALTA.** Esta opción nos proporciona la lectura en voz alta de un texto seleccionado por el usuario. Para ello, hacemos uso de la API que nos proporciona Google Chrome, realizando la siguiente llamada:

```
1 chrome.tts.speak(texto_seleccionado);
```

Además, nuestra extensión proporciona enlaces a las siguientes páginas:

- **BLOG**. Escogiendo esta opción de la aplicación el usuario podrá navegar al blog del colegio Estudio 3 AFANIAS en una nueva ventana. Esta opción es configurable, pudiendo añadir un nuevo nombre y dirección url a la que navegar. También es posible ocultarlo en caso de no necesitarlo.
- **YOUTUBE**. Con esta opción el usuario accederá a Youtube en una nueva ventana. En caso de haber realizado la búsqueda de alguna palabra, obtendrá acceso a la página con la búsqueda realizada.
- **WIKIPEDIA**. Esta opción permite al usuario acceder a la Wikipedia en una ventana nueva. Si el usuario ha seleccionado alguna palabra, accederá a la página con la búsqueda realizada.

#### 7.2.4. Toolbar

En esta carpeta generamos el fichero *createCustomToolbar.js*. Este es el archivo con el que se comunica *loadToolbarMenu.js* y su función es generar nuestra barra de menú personalizada.

En primer lugar, creamos un nuevo componente *div* cuyo identificador será *customToolbarMenu*. Posteriormente, añadiremos en este componente la vista *toolbar.html*, así, cargamos nuestra barra de menú de inicio sin personalizar por el usuario.

A continuación, añadimos la funcionalidad encargada de escuchar los cambios introducidos por el usuario para adaptar la barra de herramientas a sus necesidades. Guardaremos en una variable el array con los elementos que seleccionó el usuario. En caso de ser vacío, la barra quedará como estaba. En otro caso, esconderemos todos los elementos con ayuda de la función *hide()* que nos proporciona JavaScript y recorreremos el array mostrando los elementos que escogió.

#### 7.2.5. Modals

En esta carpeta guardaremos los ficheros encargados de generar las ventanas emergentes como resultado de alguna búsqueda por el usuario o mostrando textos informativos.

Se trata de ficheros JavaScript que contienen una función en la que generaremos el código HTML a partir de los datos recibidos por la llamada según la opción seleccionada por el usuario. Dependiendo de la información, creamos los siguientes ficheros:

- TEXTMODAL.JS: Este es el fichero encargado de generar el modal de texto correspondiente a las opciones *definiciones*, *sinónimos* y *antónimos*. En la ventana emergente nos encontraremos con la posibilidad de seleccionar los elementos. Por ejemplo, si llamamos al servicio de definiciones, el usuario podrá seleccionar las acepciones que desee guardar pulsando el botón *Guardar*.
- IMGMODAL.JS: Este fichero genera el modal correspondiente a la opción de los *pictogramas*. En la ventana emergente la aplicación mostrará una serie de imágenes haciendo referencia a la palabra escogida por el usuario, quién tendrá la posibilidad de guardar su búsqueda pulsando el botón *Guardar*.
- SUMMARYMODAL.JS: Este fichero genera el modal relacionado con la opción de *resumen*. En la ventana emergente la aplicación proporcionará al usuario el resumen del texto escogido por el usuario, quién podrá escoger guardarlo pulsando el botón *Guardar*.
- SAVECHANGESMODAL.JS: Este fichero genera el modal tras pulsar en la opción de los engranajes en la barra de menú. En la ventana emergente el usuario podrá ver todas sus búsquedas guardadas. Si lo desea, el usuario o el tutor podrán escoger la opción *Exportar* para descargar en un fichero la información guardada.
- ERRORMODAL.JS: Este fichero genera el modal tras pulsar la opción de *definiciones*, *sinónimos*, *antónimos*, *pictogramas* o *resumen* en el caso de que el usuario no haya seleccionado o buscado ninguna palabra o texto, mostrando al usuario el correspondiente mensaje informativo.
- NORESULTSSERVICEMODAL.JS: Este fichero genera el modal tras pulsar la opción de *definiciones*, *sinónimos*, *antónimos* o *pictogramas* en caso de que la palabra buscada por el usuario no exista o el servicio correspondiente no ofrezca resultados, mostrando al usuario el correspondiente mensaje informativo.

### 7.2.6. Storage

En esta carpeta almacenaremos los ficheros relacionados con la gestión de los datos guardados por el usuario. En ella podemos encontrar dos ficheros:

- EXPORTDATA.JS: Se trata de un fichero JavaScript encargado de exportar las búsquedas que el usuario ha realizado generando un archivo *.html* que el usuario se descargará. Para ello, obtenemos los datos guardados por el usuario a través de la función *chrome.storage.sync.get* proporcionada por la API de Chrome. Posteriormente, almacenaremos



los datos guardados en un *string* con formato HTML. Finalmente, generaremos un fichero con el nombre *misDatosGuardados.html* que contendrá las búsquedas guardadas del usuario, con la fecha y la hora.

- `SAVECHANGES.JS`: Se trata de un fichero JavaScript encargado de guardar los datos de las búsquedas realizadas por el usuario, haciendo uso de las funciones `chrome.storage.sync.get` y `chrome.storage.sync.set` que nos proporciona la API de Chrome. En primer lugar, guardaremos los datos que el usuario desea guardar en un array. Posteriormente, haremos uso de `chrome.storage.sync.get` para obtener datos guardados por el usuario anteriormente. En caso de ser la primera vez que el usuario guarda los datos, haremos uso `chrome.storage.sync.set` guardando los datos del array. Si el usuario ya tenía datos guardados, concatenaremos los datos de entrada almacenados en el array con los datos ya guardados, evitando sobrescribir los datos.

Además, los datos guardados por el usuario son persistentes, es decir, los datos permanecerán guardados aunque el usuario haya cerrado el navegador.

## 7.3. Instalación del plugin

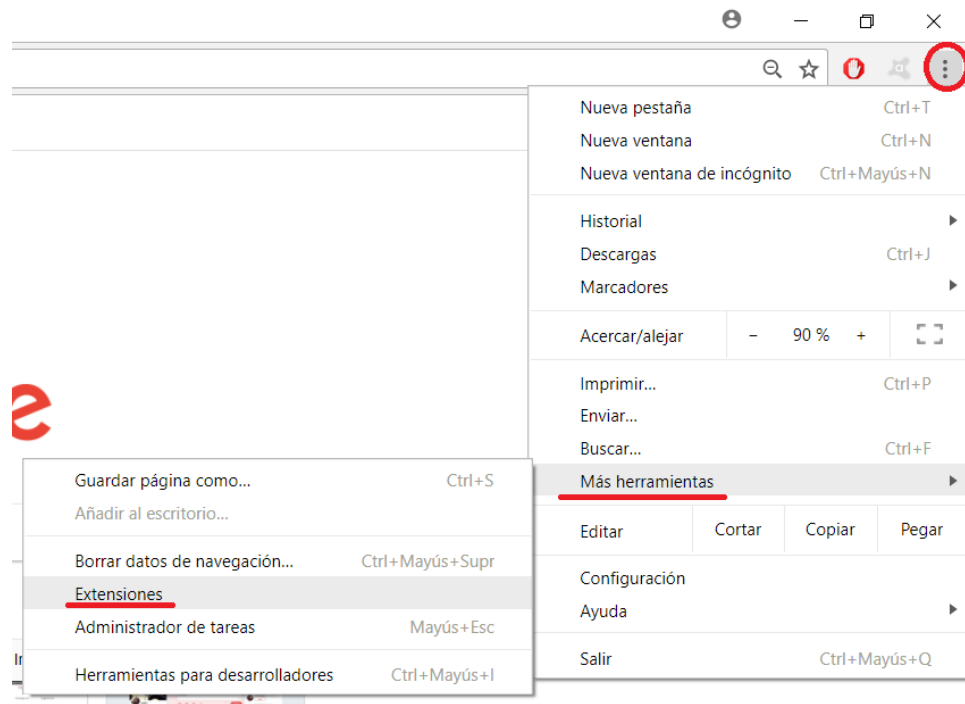
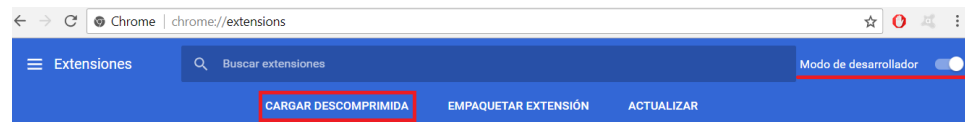
El primer paso consiste en descargar la carpeta que contiene la funcionalidad de la extensión, ubicada en el siguiente enlace:

<https://github.com/NILGroup/TFG-1718-AccesibilidadWeb>.

Una vez descargada y descomprimida la carpeta es momento de cargar nuestra aplicación en local. Para ello, entramos en Google Chrome y en la parte de arriba a la derecha del navegador, pulsamos los *tres puntos*, seleccionamos la opción *Más herramientas* seguido de *Extensiones*, como se muestra en la figura 7.2.

Una vez situados, es preciso habilitar la opción *Modo desarrollador* que el navegador de Google Chrome nos ofrece, que se encuentra en la parte superior derecha de la pantalla. Tras habilitar dicha opción, se mostrarán tres nuevas opciones como mostramos en la figura 7.3, de las cuales debemos seleccionar la opción *Cargar descomprimida*.

A continuación, se nos abrirá un popup donde debemos añadir la ruta en la que se encuentra nuestra carpeta descargada. Añadimos la ruta y pulsamos *Aceptar*. Aparecerá en la pantalla nuestra aplicación cargada desde local en el ordenador del usuario.

Figura 7.2: Ruta para acceder a la pestaña *Extensiones* del navegadorFigura 7.3: Opciones que aparecen tras habilitar el *Modo desarrollador*

## 7.4. Extensibilidad del plugin

Una vez explicado el contenido y la funcionalidad de nuestra extensión para Google Chrome, vamos a continuar explicando cómo un programador puede añadir un nuevo servicio que precise siguiendo unos pasos. De esta manera, nuestra aplicación será extensible para aquellas personas que sigan estos pasos. Es importante destacar que sus cambios solo serán visibles desde local, es decir, desde el lugar donde carguen la extensión.

Una vez cargada nuestra aplicación, debemos abrir en el entorno que deseemos el proyecto, *Atom* por ejemplo. Una vez abierto, localizamos los ficheros que se deben modificar para añadir el nuevo servicio:

— *creatingServicesMenu.js*

- *toolbar.html*
- *option-selected.js*

El primer fichero que modificaremos será *creatingServicesMenu.js*, que encontraremos en la ruta “readIt/js/selectionServicesMenu/”. En él, podemos observar la variable *features*, que contiene todos los servicios de los que dispone el plugin. El desarrollador deberá añadir ahí el par [*Clave*, *Valor*] del servicio que quiere, donde la *clave* se corresponde con el identificador con el que se guardará el servicio en la barra de herramientas y *valor* representa el nombre con el que se mostrará el servicio en nuestra barra.

Como mostramos en el siguiente ejemplo, el desarrollador deberá añadir el correspondiente [*clave*, *valor*] de su servicio descritos en la línea 10.

```

1 var features = {
2   "Definitions": "Definiciones",
3   "Synonyms": "Palabras parecidas",
4   "Antonyms": "Palabras contrarias",
5   "Youtube": "Youtube",
6   "Wikipedia": "Wikipedia",
7   "Pictograms": "Pictogramas",
8   "Summary": "Resumen",
9   "OutLoudReading": "Lectura en voz alta",
10  "Clave": "Valor"
11 };

```

El segundo archivo a retocar será *toolbar.html*, que encontramos en la ruta “readIt/”. El programador deberá añadir una línea de código para que su nuevo servicio se muestre en la barra de herramientas.

Como mostramos a continuación, el desarrollador deberá añadir su [*clave*, *valor*] como se muestra en la línea 12 del ejemplo.

```

1 <ul>
2 <li><input id="searchText" type="text" placeholder="Buscar..."
3 ></li>
4 <li id="Blog"><a href="#">Blog</a></li>
5 <li data-toggle="modal" data-target="#exampleModalLong" id="
6   Definitions" class="hidableItems"><a href="#">Definiciones
7 </a></li>
8 <li data-toggle="modal" data-target="#exampleModalLong" id="
9   Synonyms" class="hidableItems"><a href="#">Palabras
10 parecidas</a></li>
11 <li data-toggle="modal" data-target="#exampleModalLong" id="
12   Antonyms" class="hidableItems"><a href="#">Palabras
13 contrarias</a></li>
14 <li data-toggle="modal" data-target="#exampleModalLong" id="
15   Pictograms" class="hidableItems"><a href="#">Pictogramas</a>
16 </li>
17 <li id="Youtube" class="hidableItems"><a href="#">Youtube</a></li>

```

```

9 <li id="Wikipedia" class="hidableItems"><a href="#">Wikipedia</
  a></li>
10 <li data-toggle="modal" data-target="#exampleModalLong" id="
  Summary" class="hidableItems"><a href="#">Resumen</a></li>
11 <li id="OutLoudReading" class="hidableItems"><a href="#">
  Lectura en voz alta</a></li>
12 <li id="Clave" class="hidableItems"><a href="#">Valor</a></li>
13 <li data-toggle="modal" data-target="#exampleModalLong" id="
  getChangesId" style="float:right"><a href="#"><i class="fa
  fa-cogs"></i></a> </li>
14 </ul>

```

El último fichero que se debe modificar será *option-selected.js*, situado en la ruta “readIt/js/services”. En este fichero se encuentran los pasos más importantes, ya que definirán el funcionamiento de su servicio.

Para ello, es importante que el desarrollador disponga de:

- Un servicio que a través de una llamada *GET* o *POST* sea capaz de obtener la información deseada así como el formato en la que la recibe.
- Debe soportar *https*, ya que sino su servicio solo funcionará en páginas *http*.

En primer lugar, el desarrollador deberá añadir la funcionalidad para que la aplicación reconozca cuando el usuario ha escogido su opción. Para ello, el desarrollador deberá añadir el código relativo a las líneas 9-17 en el siguiente ejemplo:

```

1 //Llamar al servicio de sinónimos
2 $("#Synonyms").click(function(){
3   if (selectionableText == ""){
4     getSearchText();
5   }
6   var url = 'https://sesat.fdi.ucm.es/tfgapi/servicios/rest/
  sinonimos/json/' + userText;
7   var serviceCalled = this.textContent;
8   callingGetWebService(url, userText, serviceCalled);
9 });
10 //Llamar al servicio nuevo del desarrollador
11 $("#Clave").click(function(){
12   if (selectionableText == ""){
13     getSearchText();
14   }
15   var url = url_servicioNuevo;
16   var serviceCalled = this.textContent;
17   callingDeveloperServiceWebService(url, userText,
    serviceCalled);
18 });

```

Destacamos:

- Línea 11. Con jQuery, comprobamos si el usuario ha escogido nuestro servicio.
- Líneas 12-14. Comprobamos que el usuario haya seleccionado o buscado alguna palabra o texto.
- Línea 15. Guardamos en la variable *url* la url del nuevo servicio que el desarrollador desea añadir. En caso de que necesite conocer el texto seleccionado por el usuario, lo encontrará en la variable global *userText*.
- Línea 16. Almacenamos en la variable *serviceCalled* el nombre del nuevo servicio que el desarrollador quiere añadir. Esto servirá para guardar el título en el modal donde se mostrarán los datos generados por el nuevo servicio.
- Línea 17. Llamamos a la función *callingDeveloperServiceWebService*, que el desarrollador deberá implementar para hacer la llamada a su servicio. Podrá escoger el nombre y los parámetros que sean necesarios.

En la función *callingDeveloperServiceWebService* nombrada anteriormente, el desarrollador ha de implementar una función que parsee los datos generados por su servicio y llamar al modal correspondiente para mostrar los resultados.

En caso de que ninguno de los modales ofrecidos por la aplicación situados en la ruta *readIt/js/modals/* se adapte a la forma en la que el desarrollador quiere mostrar sus datos, deberá crear un nuevo fichero, llamado por ejemplo *developerModal.js* en la ruta anteriormente nombrada que contendrá la función *openDeveloperModal(...)*.

A continuación, mostramos un ejemplo del modal que muestra los resultados de los servicios *definiciones*, *sinónimos* y *antónimos* para explicar la estructura que ha de seguir:

```
1 function openDeveloperModal(array, serviceCalled, selectedText)
2 {
3   jQuery('#modalTitleId').empty();
4   jQuery('#modalTitleId').append(serviceCalled + ': ' +
5     selectedText);
6   jQuery('#modalBodyId').empty();
7   jQuery.each(array, function (index, value) {
8     jQuery('#modalBodyId').append('<label><input type="checkbox"
9       class="optionsSelected" id="' + index + '" value="' +
10        value + '>' + value);
11   });
12   jQuery('#saveChangesId').show();
```

```
12   jQuery('#exportChangesId').hide();  
13 }
```

Donde destacamos:

- Línea 1. El desarrollador puede añadir los atributos que necesite en su función. En el ejemplo, la variable *array* guarda, por ejemplo, las acepciones de la palabra buscada por el usuario en caso de haber solicitado las *definiciones*. La variable *serviceCalled* indica el nombre del servicio que el usuario ha solicitado. La variable *selectedText* se corresponde con el texto que el usuario ha seleccionada.
- Líneas 2 y 3. Se corresponden con el título que mostrará el modal. El desarrollador tiene libertad para generar ese formato u otro que desee para el título. En el ejemplo, mostraría el nombre del servicio solicitado seguido del texto seleccionado por el usuario.
- Línea 4. El desarrollador debe escribir esta línea, ya que sirve para limpiar el cuerpo del modal y dejarlo preparado para los datos nuevos que introducirá.
- Líneas 6-9. Se corresponden con la manera en la que se muestran los datos obtenidos del nuevo servicio. El desarrollador dispone de libertad para generar los datos de la manera que desee.
- Línea 11. El desarrollador podrá escribir esta línea tal cual viene en el ejemplo en caso de querer permitir al usuario guardar su búsqueda. En caso contrario, deberá escribir *.hide()* en lugar de *.show()*
- Línea 12. El desarrollador debe escribir esta línea, ya que no queremos mostrar el botón de exportar datos en este modal.

Siguiendo estas pautas, el desarrollador podrá incluir en la aplicación todos los servicios que precise, pudiéndolo adaptar a las necesidades de los usuarios.

## Capítulo 8

# Conclusiones y trabajo futuro

Una vez finalizado este proyecto llegamos a una serie de conclusiones y recopilamos el trabajo que esperamos desarrollar en un futuro.

### 8.1. Conclusiones

Al inicio del proyecto realizamos una etapa de aprendizaje acerca de las distintas necesidades que pueden presentar determinadas personas navegando en la web. Aprendimos que existen distintos colectivos, por ejemplo, personas que presentan discapacidad o avanzada edad, entre otros, que se enfrentan a una barrera que les dificulta el acceso a internet, encontrándose con problemas para leer, escribir o entender textos a la hora de navegar por la web.

Tras este proceso de aprendizaje descubrimos que las capacidades de comprensión lectora varían de unas personas a otras, por lo que comprendimos la necesidad y la importancia de desarrollar nuestra aplicación decidiendo cuáles serían las funcionalidades que más ayuda proporcionaría a dichos colectivos. Ante esta problemática, contamos con la ayuda de los profesionales del colegio Estudio 3 AFANIAS para escoger las funcionalidades que más se adaptaran a los usuarios.

Para desarrollar nuestra aplicación, decidimos implementar una extensión para Google Chrome que, a través de servicios externos, ofrece distintas opciones que ayudan al usuario a comprender el texto al que se enfrenta. Además, ReadIt presenta grandes ventajas:

- Permite configurar a los usuarios los servicios que necesiten, pudiendo descartar aquellos que no le ofrezcan ningún tipo de utilidad.
- Al tratarse de una extensión, está disponible para todos los usuarios que quieran usarla.

- En caso de necesitarlo, se pueden añadir nuevos servicios que no contenga ReadIt. Un desarrollador podrá incluir en la aplicación un servicio nuevo que los usuarios precisen.

Tras evaluar nuestra aplicación en el colegio Estudio 3 AFANIAS con profesores y diez alumnos del colegio, comprobamos que ReadIt no solo facilita la comprensión de los textos que ofrece internet, sino que también impulsa y anima a los usuarios a mejorar en su aprendizaje e independencia.

Hemos querido definir ReadIt como una aplicación que se configura como apoyo o ayuda para colectivos de personas que presentan algún tipo de dificultad a la hora de comprender el lenguaje escrito.

## 8.2. Trabajo futuro

ReadIt ofrece un amplio abanico de posibilidades para seguir ampliando sus funcionalidades:

- Mejorar algunos de los servicios que ofrece ReadIt, ya que algunos de ellos no soportan las tildes o caracteres especiales. Además, el servicio de antónimos rara vez ofrece resultados. Otro ejemplo sería mostrar las acepciones de las definiciones de manera más sencilla.
- Ofrecer más servicios que puedan ser de utilidad para los usuarios. Aumentar el abanico de posibilidades que ofrece ReadIt lo haría más completo y útil, ya que se podría abarcar un mayor número de problemas que presenten los usuarios a la hora de comprender el texto.
- Durante la evaluación de ReadIt con los alumnos del colegio Estudio 3 AFANIAS, uno de los profesionales nos explicó que realizar un servicio que permitiera al usuario transmitir por voz y que la aplicación recogiera los datos resultaría interesante.
- Durante esta evaluación, descubrimos que en el colegio algunos de los enlaces con los que estudian son PDF. Una propuesta interesante sería poder ofrecer los servicios que presenta nuestra aplicación en los documentos PDF.
- Una vez que el usuario busca, por ejemplo, el resumen de un texto, resultaría interesante que pudiera seleccionar las palabras que le presenten dificultades en su lectura.

Además, tras comprobar la ayuda que proporcionó nuestra aplicación en el colegio Estudio 3 AFANIAS, queremos que ReadIt llegue a más colegios



---

u otros centros para poder así eliminar en medida de lo posible esta barrera a la que se enfrentan los usuarios. Para ello, se publicará en la “Web store” de Chrome.



## Capítulo 9

# Conclusions and future work

Once this project was completed, we came to a series of conclusions and thought about the work we hope to carry out in the future.

### 9.1. Conclusions

At the beginning of the project we carried out a learning stage about the different needs that certain people may have while browsing the web. We learned that there are different groups, for example, people with disabilities or old age, among others, who face a barrier that makes it difficult for them to access the Internet, and who find it difficult to read, write or understand texts when browsing the web.

After this learning process we discovered that reading comprehension skills vary from one person to another, so we understood the need and importance of developing our application and decided which functionalities would provide the most help to these groups. Faced with this problem, we had the help of professionals at Estudio 3 AFANIAS school to choose the features that best suited the users.

To develop our application, we decided to implement an extension for Google Chrome that, through external services, offers different options to help users understand the text they are trying to read. ReadIt has great also advantages:

- Allows users to configure the services they need, and can discard those that do not offer any kind of utility for a specific user.
- As it is an extension, it is available for all users who want to use it.
- If needed, new services can be added to ReadIt. Any developer may include a new service that the users need in the application.

After evaluating our application in the Estudio 3 AFANIAS school with teachers and ten students from the school, we found that ReadIt not only makes it easier to understand the texts offered by the Internet, but also encourages users to improve their learning and independence.

We wanted to define ReadIt as an application that is configured to support or help groups of people who have some kind of difficulty understanding written language.

## 9.2. Future work

ReadIt offers a wide range of possibilities to further expand its functionality in the future:

- Improve some of the services offered by ReadIt, as some of them do not support accents or special characters. In addition, the antonyms service rarely delivers results. Another example would be to show the meanings of the definitions in a simpler way.
- Offer more services that may be useful to users. Increasing the range of possibilities offered by ReadIt would make it more complete and useful, as it could cover a greater number of problems encountered by users in understanding the text.
- During the evaluation of ReadIt with the students of the Estudio 3 AFANIAS school, one of the professionals explained to us that it would be interesting to provide a service that allowed voice input by the user.
- During this evaluation, we discovered that some of the links the students used in the center are PDF. An interesting proposal would be to be able to offer the services that our application presents in PDF documents.
- Once the user has searched, for example, for a summary of a text, it would be interesting if they could select the words that are difficult for them to read.

In addition, after realizing the help provided by our application to the Estudio 3 AFANIAS school, we want ReadIt to reach more schools or other centres in order to remove as much as possible of this barrier faced by users. Thus, it will be published in Chrome's "Web store".

# Bibliografía

- BAUTISTA, S., HERVÁS, R., GERVÁS, P., BAGÓ, A. y GARCÍA, J. Taking text simplification to the user: integrating automated modules into a web browser. En *In Proceedings of the 8th International Conference on Software Development and Technologies for Enhancing Accessibility and Fighting Info-exclusion. DSAI*. Aristotle University of Thessaloniki, Greece., 2018.
- CHAUVIN, S. Google Chrome: Análisis de sus características principales. Disponible en <http://www.mujeresdeempresa.com/google-chrome-analisis-de-sus-caracteristicas-principales/>.
- CHISHOLM, W., SLATIN, J., WHITE, J., CALDWELL, B., COOPER, M., REID, L. G. y VANDERHEIDEN, G. Web content accessibility guidelines (WCAG) 2.0. Disponible en <https://www.w3.org/TR/WCAG20/#perceivable>.
- CHISHOLM, W., VANDERHEIDEN, M. G. y JACOBS, M. I. Web content accessibility guidelines 1.0. Disponible en <https://www.w3.org/TR/WCAG10/>.
- DKEN2302. Características, ventajas y desventajas de google chrome. Disponible en <https://jditic92.wordpress.com/2015/06/27/caracteristicas-ventajas-y-desventajas-de-google-chrome/>.
- HENRY, S. L. Introduction to web accessibility. Disponible en <https://www.w3.org/WAI/fundamentals/accessibility-intro/>.
- LEONARDO, C. Historia. Disponible en <https://sites.google.com/site/navegador012/home/datos-institucionales>.
- MORA, S. L. Qué es la accesibilidad web. Disponible en <http://accesibilidadweb.dlsi.ua.es/>.
- MORENO, L. y FRANCO, C. Accesibilidad en la información: Lectura fácil. Disponible en <http://www.cesya.es/sites/default/files/documentos/AccesibilidadComunicacionLecturaFacil.pdf>.

VELASCO, R. Diferencias entre las 4 versiones de google chrome. Disponible en <https://www.redeszone.net/2017/04/24/diferencias-versiones-google-chrome/>.

W3C, M. Objetivos del W3C. Disponible en <https://www.w3c.es/Consortio/mision>.